Fitting Generalized Zero-Inflated Poisson Regression Mixture Models to Bacteria Microbiome Data

by

Siyu Chen

A Thesis

presented to

The University of Guelph

In partial fulfilment of requirements

for the degree of

Master of Science

in

Statistics

Guelph, Ontario, Canada

ABSTRACT

FITTING GENERALIZED ZERO-INFLATED POISSON REGRESSION MIXTURE MODELS TO BACTERIA MICROBIOME DATA

Siyu Chen

University of Guelph, 2018

Advisor:

Dr. Zeny Feng

Gut microbial dysbiosis contributes to the risk of colorectal cancer, thus it is important to study the gut mucosal microbiome. Gut bacteria microbiome data has the features of excess zeros and overdispersion that restrict the use of fitting traditional Poisson regression models to this kind of count data. We propose the use of the generalized zero-inflated Poisson (GZIP) regression mixture model for analyzing such data. When fitting a mixture model, we need to specify the number of components in a given population. However, the number of components is unknown. In this thesis, the Bayesian information criterion (BIC) is used to identify a preferred model with a pre-specified number of components. The EM algorithm is used to estimate parameters and the performance of the models is assessed by simulation studies. The GZIP mixture model is applied to gut bacteria microbiome data from a colorectal cancer study. We only consider the carcinoma and healthy groups as a health state covariate and select the best fitted GZIP model to each bacteria genus from models of two, three, or four components. Some special cases where the proposed methods failed to be applied are also discussed.

Acknowledgements

Firstly, I would like to thank Dr. Zeny Feng who gave me the opportunity to study in University of Guelph and great support through both my study period and thesis period. Also, I would like to thank Dr. Gerarda Darlington as my advisory committee. I could not have finished my thesis without the help of you. I would also like to thank the entire Math and Statistics department for giving me so much help and support. It is a pleasant and unforgettable memory when studying here.

Secondly, I would like to thank my parents who always support me and give me the strength to keep doing what I like to do. I would also like to thank all my friends. The encourage and urge of you are the treasures for me. I love you all.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Microbiome Data

A microbiome comprises the microorganisms or the genomes of resident microorganisms that live in one environmental niche (Bäckhed et al., 2005; Ley et al., 2006; Turnbaugh et al., 2007). The human gut microbiota are the microorganisms that reside on or within the human gastrointestinal (GI) tract. Antonie van Leewenhoek initiated studies of human microbiome diversity in the 1680s (Ursell et al., 2012). Now, with the rapid advancement in biological technology, two sequencing techniques are often used to identify and characterize bacteria in a given sample. The first one is the targeted amplicon sequencing technique (Bybee et al., 2011). This technique sequences a common gene, such as the 16S ribosome RNA (rRNA) gene, that exists in all bacteria and contains both slowly evolving regions and fast evolving regions. The variation of the sequences in these regions is able to distinguish bacteria from species to species. This technique is very popular for studies of bacteria but has limitations for studies of other microorganisms such as viruses which do not have ribosomes. The focus of this thesis will be on bacteria microbiome data only. The second sequencing technique is metagenomic, particularly the shotgun sequencing technique (Gardner et al., 1981; Doc-

trow, 2016), which directly recovers all genes from the whole genome of all microorganisms from given samples. Although some initial steps for these two approaches are similar, for instance demultiplexing (Ursell et al., 2012), there are some differences between them. For example, for the 16S rRNA pipeline, sequences must be set into operational taxonomic units (OTUs) while for the shotgun metagenomics, sequences must be allocated into functions and taxonomy. OTUs are the most commonly used units for bacterial microbiome diversity. They can be clustered into groups according to the similarity of the sequences. Based on the similarity threshold which is usually 97% (Grice et al., 2008), OTUs can be defined. Within each group, a sequence that is different from others will be used for bacteria identification by screen through a database of known bacterial sequences. By doing this, the bacterium is identified and is quantified by the count of these similar sequences. However, the 16S rRNA sequencing technique is limited to bacteria identification and quantification only. To study the whole genome of a microorganism, the metagenomic shotgun sequencing technique is used.

Bacteria microbiome data has several typical features. Firstly, OTUs are count data. Then, there is a taxonomic rank for bacteria: Domain, Kingdom, Phylum, Class, Order, Family, Genus and Species. Since many bacteria have very similar sequences at the species level such that some of them are only different in one or a few nucleotides, it is difficult to infer whether two sequences are representing two different species or are from the same species but have sequencing errors that makes them different. For this reason, most studies will stop the taxonomic assignment at the genus level and will not go further to the species level. Thus, in our research, we analysed the OTU bacteria data at the genus level instead of the species level. Thirdly, microbiome data is often sparse comprising of many zero counts. This phenomenon is typically referred to as zero-inflated counts which means that the count data has an excess of zero counts compared to what is expected under some distributional assumption such as a Poisson distribution. In addition, the OTU data often

exhibits overdispersion with a variance typically larger than the mean.

More detailed introductions of two sequencing techniques and human gut microbiome data are in Chapter 2.

1.2 GZIP Regression Mixture Model

As the Poisson distribution is a common choice for analyzing count data, it is considered for the OTU count data. However, the Poisson distribution for count data may have limitations in the presence of excess zeros and overdispersion. To overcome the problem of excess zeros in count data, Lambert (1992) proposed the zero-inflated Poisson (ZIP) regression model and demonstrated its application in manufacturing problems. In this model, the population of zero counts is divided into two subpopulations, one is a population having a Poisson distribution and the other is a population of zeros. The ZIP regression model has a mixing weight parameter π for the proportion of the count data from the Poisson distribution and $(1 - \pi)$ is the proportion of count data from the population of zeros. In addition, the Poisson distribution mean depends on covariates through a regression model. The detail of the ZIP regression model will be discussed in Section 3.1.1.

To find the maximum likelihood estimates for mixture models like ZIP models, the expectation-maximization (EM) algorithm is usually used. The EM algorithm was first proposed by Dempster, Laird, and Rubin (1977). It is a widely applicable iterative method for finding maximum likelihood estimates and is best used in incomplete-data problems. There are two steps, the expectation step (E-step) and the maximization step (M-step), in each iteration of an EM algorithm. In this thesis, the EM algorithm is used to estimate the parameters in the ZIP model. The ZIP regression model only handles excess zero data, but the overdispersion problem is not handled by the Poisson distribution. In this thesis, we propose to use the generalized ZIP (GZIP) regression mixture model (Lim et al., 2014) to

handle overdispersion. The term "*generalized*" allows the mixture model to consist of more than two components. For example, a three components mixture, one for population of zeros and two different Poisson distributions, could be used to fit the count data. The model also allows linking covariates to the Poisson mean parameter of each component as well as the mixing proportion of each component.

1.3 Objective and Structure of the Thesis

The objective of this thesis is to investigate the possibility and suitability of using the GZIP regression mixture model to model the distribution of the microbiome data. If the GZIP regression mixture model is an appropriate tool, an association test for the association between the disease status and the bacteria can be established based on this GZIP regression mixture model.

This thesis is organized as follows: in Chapter 2, we introduce more detail regarding microbiome data and in Chapter 3, the methodologies used in this thesis, which include the GZIP regression mixture model and the EM algorithm, will be described. Simulation studies evaluating model performance are in Chapter 4. Chapter 5 demonstrates the application of the GZIP regression mixture model to real data. Finally, Chapter 6 provides discussion and future work.

Chapter 2

Human Gut Microbiome Data

The human colon is known to be a densely populated microbial community containing bacteria, archaea, fungi, and viruses (Sartor and Wu, 2017). Microbes do not live alone. Instead, they are commensal, interacting with others, competing for nutrients, and functioning integratedly and thereby affecting their living environment and ecosystem. The human gut microbiome refers to the collective genomes of microbes in the human gut. Carding et al. (2015) pointed out that the composition of the human gut microbiome is largely affected by diet, toxins, drugs, and pathogens. On the other hand, gut microbial dysbiosis (microbial imbalance or maladaptation) contributes to the development of colonic diseases such as colorectal cancer (carcinoma) (Tamboli et al., 2004; Moos et al., 2016).

Determining bacterial population dynamics is a crucial first step for studying the association between microbial environment and health or other outcomes. Currently, two approaches based on next generation sequencing (NGS) techniques, the targeted gene amplicon and the shotgun metagenomics, are often used for characterizing and identifying bacteria in a given sample. Sequencing the targeted gene, for example, the 16s rRNA gene, has been the mainstay for bacteria microbiome studies where the main purposes are to identify and quantify the bacteria in a given sample. This approach utilizes the fact that the 16s rRNA

gene is ubiquitous in bacteria and contains nine hypervariable regions (V1-V9) that serve as a markers to distinguish bacteria from species to species (Kumar et al., 2011). However, the 16s rRNA sequencing data do not provide information regarding bacterial functionality and bacterial genetic information. In addition, the identification of bacteria at the species level is difficult and thus most bacteria microbiome studies classify bacteria at the lower level genus rank (Janssen, 2006; Yarza et al., 2014).

The shotgun sequencing approach is used to sequence all bacterial microbial genomic DNA and thus provides a great opportunity to characterize microbial communities and to study the functionality of genes coded by these bacteria. Shotgun sequencing enables identification of bacteria at the finer rank of species level. However, this approach poses many computational and statistical challenges as it generates much more data that makes the sequencing alignment and bacteria identification much harder and more computationally costly.

In this thesis, we will apply the method of Lim et al. (2014) to analyse the gut bacteria microbiome data from the colorectal cancer study conducted by Nakatsu et al. (2015). The study collected tissue samples from gut mucosa from 160 subjects. Of those, 61 subjects were healthy independent subjects, 47 were patients with histology-verified adenoma, and 52 were patients with invasive adenocarcinoma (Nakatsu et al., 2015). The study aimed to investigate the differences in the composition of gut microbiome communities at different stages of colorectal tumorigenesis. The bacteria microbiome data were prepared using the 16s rRNA gene sequencing technique. The raw read sequencing data is downloaded from the Sequence Read Archive (SRA) –National Center for Biotechnology Information (NCBI) database. With the raw read sequence data, a computational pipeline based on the "Mothur" software package (Schloss et al., 2009) is prepared by Bak (2017) for identifying and quantifying the bacteria abundance in each sample and then generate the final dataset used in this thesis. The pipeline consists of the following steps: aligning sequences to reference genome,

assembling sequences to contig which is a set of overlapping DNA segments and all of them together express a consensus region of DNA (Gregory, 2005), clustering contigs to OTUs, and finally assigning taxonomies to OTUs. Bacteria are classified into taxonomic ranks from the highest level of species, and then in turn, are in lower hierarchical ranks: genus, family, order, class, phylum, kingdom and domain. The "Mothur" software allows a user to specify the highest taxonomy. It is known that when using the 16S gene, bacteria at the species level often differ by 1 or a few nucleotides. With contigs that are different by 1 or a few nucleotides, it is difficult to distinguish whether the contigs represent different species or are from the same species but have different sequences due to sequencing errors. So, we specify the highest taxonomic rank to be the genus level. The OTU counts are summarized in an OTU table where each row corresponds to a sample and each column provides the count of bacteria for each genus.

In this thesis, we only focus on the comparison between the invasive adenocarcinoma patient (case) group and the healthy subject group (control). For each bacteria genus, we use the GZIP regression mixture model to fit the data, where the case and control status is the covariate included in the model and the OTU count of the bacteria is the response.

Chapter 3

Methodology

3.1 Generalized Zero-Inflated Poisson Regression Mixture Model

In a K components GZIP mixture model, the count response variable, Y, is assumed to follow the distribution function given as (Lim et al., 2014):

$$P(Y = y|\boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^{K-1} \pi_k f_p(y; \theta_k) + (1 - \sum_{k=1}^{K-1} \pi_k) I_{[y=0]} \tag{3.1}$$

where $f_p(y; \theta_k) = \frac{e^{-\theta_k}\theta_k^y}{y!}, y = 0, 1, 2, \cdots$ and k=1, 2, $\cdots$, K. $I_{[*]}$ is the indicator function such that $I_{[*]} = 1$ if the statement denoted by " $*$ " is true or $I_{[*]} = 0$ otherwise. Alternatively, the distribution function can be rewritten as (Lim et al., 2014):

$$P(Y = y|\boldsymbol{\pi}, \boldsymbol{\theta}) = \begin{cases} \pi_1 e^{-\theta_1} + \pi_2 e^{-\theta_2} + \cdots + \pi_{K-1} e^{-\theta_{K-1}} + \pi_K, & y = 0 \\ \frac{\pi_1 e^{-\theta_1}\theta_1^y}{y!} + \frac{\pi_2 e^{-\theta_2}\theta_2^y}{y!} + \cdots + \frac{\pi_{K-1} e^{-\theta_{K-1}}\theta_{K-1}^y}{y!}, & y > 0. \end{cases}$$

Here, the mixing weight parameter vector is given by $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_K)^T$ with $\pi_k \geq 0$,

for $k = 1, \cdots, K$, and $\sum_{k=1}^{K} \pi_k = 1$. Thus, we can reduce the vector $\boldsymbol{\pi}$ to $(K-1)$ elements such that $\boldsymbol{\pi} = (\pi_1, \cdots, \pi_{K-1})^T$ and $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$. The vector for the mean of each Poisson distribution with the probability mass function $f_p(y; \theta_k)$ is $\boldsymbol{\theta} = (\theta_1, \cdots, \theta_{K-1})^T$. In many situations, the mixing weights, $\boldsymbol{\pi}$, and the means, $\boldsymbol{\theta}$, depend on covariates $\boldsymbol{x}_i = (1, x_{i1}, \cdots, x_{ip})$, where p is the number of covariates for each subject i, $i = 1, \cdots, n$. A multinomial regression model with a logit link function is used to link $\boldsymbol{x}_i$ to π_{ik} and a regression model with a log-link function is used to link $\boldsymbol{x}_i$ to θ_{ik} as follows:

$$\begin{cases} \pi_{ik} = \frac{e^{\boldsymbol{x}_i \boldsymbol{\beta}_k}}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{x}_i \boldsymbol{\beta}_k}} \\ \pi_{iK} = 1 - \sum_{k=1}^{K-1} \pi_{ik} = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\boldsymbol{x}_i \boldsymbol{\beta}_k}} \end{cases}$$

where $k = 1, \cdots, K-1$, and

$$\theta_{ik} = e^{\boldsymbol{x}_i \boldsymbol{\gamma}_k}. \tag{3.2}$$

In π_{ik} and θ_{ik}, if there are p covariates to be considered, we have $\boldsymbol{x}_i = (1, x_{i1}, \cdots, x_{ip})$, $\boldsymbol{\beta}_k = (\beta_{0k}, \beta_{1k}, \cdots, \beta_{pk})^T$ and $\boldsymbol{\gamma}_k = (\gamma_{0k}, \gamma_{1k}, \cdots, \gamma_{pk})^T$. Note, because $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ now depend on covariates $\boldsymbol{x}_i$ for each subject, they become subject-specific and $\boldsymbol{\pi}_i$ and $\boldsymbol{\theta}_i$ are no longer constant for all subjects. The model in Eq. (3.1) can be reparameterized in terms of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Thus, the K components GZIP regression mixture model becomes (Lim et al., 2014):

$$f(y_i; \boldsymbol{\beta}, \boldsymbol{\gamma}, x_i) = \sum_{k=1}^{K-1} \pi_k(\boldsymbol{\beta}_k, x_i) f_p(y_i; \boldsymbol{\gamma}_k, x_i) + (1 - \sum_{k=1}^{K-1} \pi_k(\boldsymbol{\beta}_k, x_i)) I_{[y_i=0]} \tag{3.3}$$

where $\boldsymbol{\beta}_k = (\beta_{0k}, \beta_{1k}, \cdots, \beta_{pk})^T$ and $\boldsymbol{\gamma}_k = (\gamma_{0k}, \gamma_{1k}, \cdots, \gamma_{pk})^T$, $\boldsymbol{\beta}_k$ and $\boldsymbol{\gamma}_k$ are two regression intercept and coefficients vectors for the k^{th} component.

In Sections 3.1.1 - 3.1.3, we will present three GZIP regression mixture models with $K = 2, 3$, and 4, respectively.

3.1.1 2 Components ZIP Regression Model

When $K = 2$, the 2 components GZIP regression mixture model becomes the ZIP regression model. The ZIP regression model has the form:

$$f(y_i; \boldsymbol{\pi}_i, \boldsymbol{\theta}_i) = \boldsymbol{\pi}_i f_p(y_i; \boldsymbol{\theta}_i) + (1 - \boldsymbol{\pi}_i) I_{[y_i=0]} \tag{3.4}$$

where $\boldsymbol{\pi}_i = \frac{e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}}}{1+e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}}}$ and $\boldsymbol{\theta}_i = e^{\gamma_0+\gamma_1 x_{i1}+\cdots+\gamma_p x_{ip}}$.

3.1.2 3 Components GZIP Regression Model

The 3 components GZIP regression mixture model is defined as:

$$f(y_i; \boldsymbol{\pi}_i, \boldsymbol{\theta}_i) = \boldsymbol{\pi}_{i1} f_p(y_i; \boldsymbol{\theta}_{i1}) + \boldsymbol{\pi}_{i2} f_p(y_i; \boldsymbol{\theta}_{i2}) + (1 - \boldsymbol{\pi}_{i1} - \boldsymbol{\pi}_{i2}) I_{[y_i=0]} \tag{3.5}$$

where $\boldsymbol{\pi}_{i1} = \frac{e^{\beta_{01}+\beta_{11} x_{i1}+\cdots+\beta_{p1} x_{ip}}}{1+\sum_{j=1}^{2} e^{\beta_{0j}+\beta_{1j} x_{i1}+\cdots+\beta_{pj} x_{ip}}}$, $\boldsymbol{\pi}_{i2} = \frac{e^{\beta_{02}+\beta_{12} x_{i1}+\beta_{22} x_{i2}+\cdots+\beta_{p2} x_{ip}}}{1+\sum_{j=1}^{2} e^{\beta_{0j}+\beta_{1j} x_{i1}+\cdots+\beta_{pj} x_{ip}}}$, $\boldsymbol{\theta}_{i1} = e^{\gamma_{01}+\gamma_{11} x_{i1}+\cdots+\gamma_{p1} x_{ip}}$ and $\boldsymbol{\theta}_{i2} = e^{\gamma_{02}+\gamma_{12} x_{i1}+\cdots+\gamma_{p2} x_{ip}}$.

3.1.3 4 Components GZIP Regression Model

Straightforwardly, the 4 components GZIP regression mixture model is defined as:

$$f(y_i; \boldsymbol{\pi}_i, \boldsymbol{\theta}_i) = \sum_{k=1}^{3} \boldsymbol{\pi}_{ik} f_p(y_i; \boldsymbol{\theta}_{ik}) + (1 - \sum_{k=1}^{3} \boldsymbol{\pi}_{ik}) I_{[y_i=0]} \tag{3.6}$$

where $\boldsymbol{\pi}_{i1} = \frac{e^{\beta_{01}+\beta_{11} x_{i1}+\cdots+\beta_{p1} x_{ip}}}{1+\sum_{j=1}^{3} e^{\beta_{0j}+\beta_{1j} x_{i1}+\cdots+\beta_{pj} x_{ip}}}$, $\boldsymbol{\pi}_{i2} = \frac{e^{\beta_{02}+\beta_{12} x_{i1}+\cdots+\beta_{p2} x_{ip}}}{1+\sum_{j=1}^{3} e^{\beta_{0j}+\beta_{1j} x_{i1}+\cdots+\beta_{pj} x_{ip}}}$, $\boldsymbol{\pi}_{i3} = \frac{e^{\beta_{03}+\beta_{13} x_{i1}+\cdots+\beta_{p3} x_{ip}}}{1+\sum_{j=1}^{3} e^{\beta_{0j}+\beta_{1j} x_{i1}+\cdots+\beta_{pj} x_{ip}}}$ and $\boldsymbol{\theta}_{i1} = e^{\gamma_{01}+\gamma_{11} x_{i1}+\cdots+\gamma_{p1} x_{ip}}$, $\boldsymbol{\theta}_{i2} = e^{\gamma_{02}+\gamma_{12} x_{i1}+\cdots+\gamma_{p2} x_{ip}}$, $\boldsymbol{\theta}_{i3} = e^{\gamma_{03}+\gamma_{13} x_{i1}+\cdots+\gamma_{p3} x_{ip}}$.

3.2 EM Algorithm For Parameter Estimation

3.2.1 Why EM algorithm

In Dempster, Laird, and Rubin (1977), the EM algorithm was used to model heterogeneous data with finite mixture models. For simplicity, let us consider a GZIP mixture model without covariates. The GZIP mixture model with K components is given by:

$$f(y_i;\boldsymbol{\pi},\boldsymbol{\theta}) = \sum_{k=1}^{K-1} \pi_k f_p(\boldsymbol{y}_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}. \tag{3.7}$$

For a random sample of n observations, the log-likelihood function for $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ is defined as:

$$\begin{aligned} l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y}) &= \log L(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y}) \\ &= \log \prod_{i=1}^{n} f(y_i;\boldsymbol{\pi},\boldsymbol{\theta}) \\ &= \sum_{i=1}^{n} \log[\sum_{k=1}^{K-1} \pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}]. \end{aligned} \tag{3.8}$$

Taking partial derivatives with respect to π_k's and θ_k's, we get a system of likelihood equations by setting these derivatives equal to zero such that:

$$\frac{\partial l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \pi_k} = \sum_{i=1}^{n} \frac{f_p(y_i;\theta_k) - I_{[y_i=0]}}{\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}} = 0 \tag{3.9}$$

$$\frac{\partial l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \theta_k} = \sum_{i=1}^{n} \frac{\pi_k \frac{\partial f_p(y_i;\theta_k)}{\partial \theta_k}}{\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}} = 0 \tag{3.10}$$

where $k = 1, \cdots, K-1$. Closed form solutions for finding the maximum likelihood estimates for parameters in the GZIP mixture model are not possible.

An iterative procedure such as the Newton-Raphson method can be considered. According to the score functions on the left hand side of Eq. (3.9) and (3.10), the second partial

derivatives with respect to π_k's and θ_k's are:

$$\frac{\partial^2 l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \pi_k^2} = \sum_{i=1}^{n} \frac{-[f_p(y_i;\theta_k) - I_{[y_i=0]}]^2}{[\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}]^2}$$

$$\frac{\partial^2 l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \pi_k \pi_k'} = \sum_{i=1}^{n} \frac{-[f_p(y_i;\theta_k) - I_{[y_i=0]}]^2}{[\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}]^2}$$

$$\frac{\partial^2 l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \theta_k^2} = \sum_{i=1}^{n} \frac{\pi_k \frac{\partial^2 f_p(y_i;\theta_k)}{\partial \theta_k^2}\Big(\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}\Big) - (\pi_k \frac{\partial f_p(y_i;\theta_k)}{\partial \theta_k})^2}{[\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}]^2}$$

$$\frac{\partial^2 l(\boldsymbol{\pi},\boldsymbol{\theta};\boldsymbol{y})}{\partial \theta_k \theta_k'} = \sum_{i=1}^{n} \frac{-\pi_k \frac{\partial f_p(y_i;\theta_k)}{\partial \theta_k} \pi_k' \frac{\partial f_p(y_i;\theta_k')}{\partial \theta_k'}}{[\sum_{k=1}^{K-1}\pi_k f_p(y_i;\theta_k) + (1-\sum_{k=1}^{K-1}\pi_k) I_{[y_i=0]}]^2}.$$

Since, for instance, $\theta^{(i+1)} = \theta^{(i)} + [H(\theta^{(i)})]^{-1} S(\theta^{(i)})$, where $H(\theta)$ is the Fisher information matrix with second partial derivatives above and $S(\theta^{(i)})$ is the score function, if the Newton-Raphson algorithm is used, the calculation of the inverse of the Fisher information matrix and score functions are needed in each iteration. When calculating the second partial derivatives in each iteration, the common denominators should be recalculated and the common denominators are subject to the sample size and the observations. Thus, Newton-Raphson method is computationally intensive. Therefore, the iterative scheme of the EM algorithm is a more effective method. Another reason why EM algorithm is used is that this method ensures the non-decreasing of the likelihood values for each iteration. When the likelihood sequences converge, the results are the MLEs.

3.2.2 EM algorithm

The EM algorithm will be used as follows as in Lim et al. (2014). If the number of components, K, is known, the likelihood function is given as:

$$L(\boldsymbol{\beta},\boldsymbol{\gamma}) = \prod_{i=1}^{n}[\sum_{k=1}^{K-1}\pi_{ik}(\boldsymbol{\beta}_k;x_i) f_p(y_i;\boldsymbol{\gamma}_k,x_i) + (1-\sum_{k=1}^{K-1}\pi_{ik}) I_{y_i=0}]. \tag{3.11}$$

This is called the incomplete-data likelihood function. Let $\boldsymbol{Z}_i = (Z_{i1}, Z_{i2}, ..., Z_{ik})^T$ be a vector of unobservable random variables indicating to which component subject i belongs such that:

$$Z_{ik} = \begin{cases} 1, & \text{if the } i^{th} \text{ subject comes from the } k^{th} \text{ component} \\ 0, & \text{otherwise.} \end{cases}$$

Note, $\boldsymbol{Z}_i$ is not observable and is assumed to follow a multinomial(1, $\boldsymbol{\pi}_i$) distribution. Combining the observed data ($\boldsymbol{y}$, $\boldsymbol{x}$) and the missing data $\boldsymbol{Z}_i$, for n subjects, the likelihood for complete data can be written as:

$$L_c(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{i=1}^{n} \Big[\prod_{k=1}^{K-1} [\pi_{ik}(\boldsymbol{\beta}_k; x_i) f_p(y_i; \boldsymbol{\gamma}_k, x_i)]^{Z_{ik}} \Big] \Big[(1 - \sum_{k=1}^{K-1} \pi_{ik}) I_{[y_i=0]} \Big]^{1-\sum_{k=1}^{K-1} Z_{ik}}. \tag{3.12}$$

The complete-data log-likelihood function becomes (Lim et al., 2014):

$$\begin{aligned} l_c(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \sum_{i=1}^{n} \Big\{ \sum_{k=1}^{K-1} Z_{ik} \log[\pi_{ik}(\boldsymbol{\beta}_k, x_i) f_p(y_i; \boldsymbol{\gamma}_k, x_i)] + \\ (1 - \sum_{k=1}^{K-1} Z_{ik}) [\log(1 - \sum_{k=1}^{K-1} \pi_{ik}(\boldsymbol{\beta}_k, x_i))] I_{[y_i=0]} \Big\} \end{aligned} \tag{3.13}$$

Given $l_c(\boldsymbol{\beta}, \boldsymbol{\gamma})$, the EM algorithm proceeds as described in the following subsection.

3.2.2.1 E-step

Suppose the EM algorithm starts with a set of initial values for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ denoted as $\boldsymbol{\beta}^{(0)}$ and $\boldsymbol{\gamma}^{(0)}$, respectively. Given the parameter values of $(\boldsymbol{\beta}^{(0)}, \boldsymbol{\gamma}^{(0)})^T$, we have $\boldsymbol{\pi}^{(0)}$ and $\boldsymbol{\theta}^{(0)}$. Then, we take the conditional expectation of $l_c(\boldsymbol{\beta}^{(0)}, \boldsymbol{\gamma}^{(0)})$ as:

$$\begin{aligned} Q &= E(l_c(\boldsymbol{\beta}^{(0)}, \boldsymbol{\gamma}^{(0)}) | \boldsymbol{y}, \boldsymbol{x}, \boldsymbol{\beta}^{(0)}, \boldsymbol{\gamma}^{(0)}) \\ &= E\Big(\sum_{i=1}^{n} \Big\{ \sum_{k=1}^{K-1} Z_{ik} \log[\pi_{ik}^{(0)} f_p(y_i; \theta_{ik}^{(0)})] + (1 - \sum_{k=1}^{K-1} Z_{ik}) \Big[\log(1 - \sum_{k=1}^{K-1} \pi_{ik}^{(0)}) \Big] I_{[y_i=0]} \Big\} \Big) \end{aligned} \tag{3.14}$$

Since in Eq (3.14), only $\boldsymbol{Z}_{ik}$ is non-constant, finding the conditional expectation of l_c simply requires the calculation of the conditional expectation of Z_{ik}. It follows that:

$$\begin{aligned} Z_{ik}^{(1)} &= E(Z_{ik}|\boldsymbol{\beta}^{(0)}, \boldsymbol{\gamma}^{(0)}, \boldsymbol{y}, \boldsymbol{x}) \\ &= \frac{\pi_{ik}^{(0)} f_p(y_i; \theta_{ik}^{(0)})}{\sum_{k=1}^{K-1} \pi_{ik}^{(0)} f_p(y_i; \theta_{ik}^{(0)}) + (1 - \sum_{k=1}^{K-1} \pi_{ik}^{(0)}) I_{[y_i=0]}} \end{aligned} \tag{3.15}$$

where $\pi_{ik}^{(0)} = \frac{e^{\boldsymbol{\beta}_k^{(0)} \boldsymbol{x}_i}}{1+\sum_{k=1}^{K-1} e^{\boldsymbol{\beta}_k^{(0)} \boldsymbol{x}_i}}$ and $\theta_{ik}^{(0)} = e^{\boldsymbol{\gamma}_k^{(0)} \boldsymbol{x}_i}$.

3.2.2.2 M-step

Given $Z_{ik}^{(1)}$, the M-step maximizes Q with respect to $(\boldsymbol{\beta}, \boldsymbol{\gamma})^T$. Suppose Q is now of the form (Lim et al., 2014):

$$\begin{aligned} Q(\boldsymbol{\beta}, \boldsymbol{\gamma}) &= \sum_{i=1}^{n} \Big\{ \sum_{k=1}^{K-1} Z_{ik}^{(1)} \log[\pi_{ik}(\boldsymbol{\beta}_k, \boldsymbol{x}_i) f_p(y_i; \boldsymbol{\gamma}_k, \boldsymbol{x}_i)] + \\ &\quad (1 - \sum_{k=1}^{K-1} Z_{ik}^{(1)})[log(1 - \sum_{k=1}^{K-1} \pi_{ik}(\boldsymbol{\beta}_k, \boldsymbol{x}_i))] I_{[y_i=0]} \Big\} \\ &= \sum_{i=1}^{n} \Big\{ \sum_{k=1}^{K-1} Z_{ik}^{(1)} \log(\pi_{ik}(\boldsymbol{\beta}_k, \boldsymbol{x}_i)) + (1 - \sum_{k=1}^{K-1} Z_{ik}^{(1)}) \log(1 - \pi_{ik}(\boldsymbol{\beta}_k, \boldsymbol{x}_i)) I_{[y_i=0]} \Big\} \\ &\quad + \sum_{i=1}^{n} [\sum_{k=1}^{K-1} Z_{ik}^{(1)} \log f_p(y_i; \boldsymbol{\gamma}_k, \boldsymbol{x}_i)] \\ &= Q_1 + Q_2 \end{aligned} \tag{3.16}$$

where Q_1 and Q_2 are the first and second terms in Eq (3.16), respectively. Since $\boldsymbol{\beta}$ is only involved in Q_1 and $\boldsymbol{\gamma}$ is only involved in Q_2, maximizing Q with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is equivalent to maximizing Q_1 with respect to $\boldsymbol{\beta}$ and maximizing Q_2 with respect to $\boldsymbol{\gamma}$. Note that Q_1 is the log-likelihood function for a multinomial logistic regression model and Q_2 is the log-likelihood function for a weighted Poisson regression model. Thus, iteratively

reweighted least squares (IRWLS) (Burrus, 2012) can be used to find the MLEs of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. Note, we use the function "*multinom*" in the R software (R Core Team, 2017) package "*nnet*" (Venables and Ripley, 2002) to find MLEs of $\boldsymbol{\beta}$ because $\boldsymbol{Z}_i$ follows a multinomial(1, $\boldsymbol{\pi}_i$) distribution and $\boldsymbol{\pi}_i$ is a function of $\boldsymbol{\beta}$. Function "*glm*" with Poisson family is used to find MLEs of $\boldsymbol{\gamma}$ within each M-step for each iteration. Then, the updated parameters are denoted as $\boldsymbol{\beta}^{(r+1)}$ and $\boldsymbol{\gamma}^{(r+1)}$ for $r = 0, 1, 2, \cdots$. For example, in the 1^{st} iteration, we obtain $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\gamma}^{(1)}$ for given $Z^{(1)}$, $\boldsymbol{y}$, and $\boldsymbol{x}$.

Given $\boldsymbol{\beta}^{(1)}$ and $\boldsymbol{\gamma}^{(1)}$, we repeat the E-step to obtain the update $Z_{ik}^{(2)}$'s and M-step to obtain the update MLEs of $\boldsymbol{\beta}^{(2)}$ and $\boldsymbol{\gamma}^{(2)}$. The EM algorithm is repeated until a prespecified stopping criterion is met.

3.2.3 Stopping Criterion

An Aitken acceleration-based stopping criterion (McLachlan and Krishnan, 2007) is used in EM algorithm to determine whether the parameter estimates have converged. In general, a stopping criterion can depend on the changes of log-likelihood values or the changes of parameter values (McLachlan and Peel, 2004). For a stopping criterion based on the difference between the previous and current parameter estimates, the iteration may stop at the location when the likelihood function reaches a local maximum instead of a global maximum. Since it is possible that more than one local maximum exists, a stopping criterion based on the results of a sequence of iterations would be a better choice.

We assume the sequence of incomplete-data log likelihood values is convergent to some value l^* (McLachlan and Peel, 2004). Then,

$$l_A^{(r+1)} = l^{(r)} + \frac{1}{(1 - c^{(r)})}(l^{(r+1)} - l^{(r)}) \tag{3.17}$$

is the Aitken accelerated estimate of l^* at the $(r+1)^{th}$ iteration. $l^{(r)}$ is the incomplete-data

log-likelihood at the r^{th} iteration and the Aitken acceleration for r^{th} iteration is:

$$c^{(r)} = \frac{l^{(r+1)} - l^{(r)}}{l^{(r)} - l^{(r-1)}}. \tag{3.18}$$

The EM algorithm will be stopped when $|l_A^{(r+1)} - l_A^{(r)}| < M$ where M is the prespecified tolerance, for example, $M = 10^{-4}$. Since $l_A^{(r+1)}$ is based on $l^{(r-1)}$, $l^{(r)}$ and $l^{(r+1)}$, and $l_A^{(r)}$ is based on $l^{(r-2)}$, $l^{(r-1)}$ and $l^{(r)}$, the Aitken acceleration-based stopping criterion stops the EM algorithm based on the log-likelihood values of 4 iterations.

3.3 Model Selection Criterion

The number of components, K, is generally unknown and, therefore, needs to be estimated. Based on pilot analyses fitting the bacteria OTU data at the genus level, most genera can be fitted by K ranging from 2 to 4 and some failed to be fitted by using the GZIP regression mixture model due to lack of convergence. A discussion of genera that failed to be fitted by the GZIP regression mixture model will be presented in Chapter 5. For the rest, we will consider K values of 2, 3, and 4 where the Bayesian information criterion (BIC) is used to determine the optimal K value.

The BIC was first introduced by Schwarz et al. (1978) and later was used by Roeder and Wasserman (1997) for mixture models. It is defined as:

$$BIC = p\log(n) - 2l(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\gamma}}; \boldsymbol{y}, \boldsymbol{x}) \tag{3.19}$$

where p is the number of parameters being estimated which here is the total number of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ parameters, n is the sample size and l is the maximized log-likelihood with the values of $\hat{\beta}$ and $\hat{\gamma}$ which are the MLEs of β and γ fitted by the EM algorithm. The model with the the smallest BIC is considered the preferred model.

3.4 Performance Assessment

The adjusted rand index (ARI) can be used to measure the accuracy of classification based on a given model relative to the true and the estimated cluster classification (Rand, 1971). The predicted component memberships at the MLEs of parameters (i.e. at the convergence of the EM algorithm) are given by $\hat{Z}_{ik}$ and subject i will be assigned to component k if $\hat{Z}_{ik}$ has the maximum values among all $\hat{Z}_{ij}$, for $j = 1, \cdots, K$. The ARI makes an adjustment to the Rand index (RI) (Rand, 1971). The RI is defined as:

$$RI = \frac{\text{Number of agreements}}{n} \tag{3.20}$$

and so $0 \leq RI \leq 1$ where 1 indicates perfect classification agreement. However, $E(RI) > 0$ under random classification. ARI is a measure to improve the RI since it can avoid agreement by chance under random classification.

Among n observations, denote V to be the predicted partition of K' groups as $V = \{V_1, \cdots, V_{K'}\}$ and compare to the true partition, T, of K groups as $T = \{T_1, \cdots, T_K\}$ where K may or may not be equal to K'. Table 3.1 and Eq (3.21) illustrate how ARI is calculated. In R, we can use the function "*adjustedRandIndex*" in the "*mclust*" package (Maechler et al., 2017) to calculate ARI.

$T \setminus V$	V_1	V_2	$\cdots$	$V_{K'}$	Total
T_1	n_{11}	n_{12}	$\cdots$	$n_{1K'}$	a_1
T_2	n_{21}	n_{22}	$\cdots$	$n_{2K'}$	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
T_K	n_{K1}	n_{K2}	$\cdots$	$n_{KK'}$	a_K
Total	b_1	b_2	$\cdots$	b_K	n

Table 3.1: ARI contingency table

The ARI is defined as:

$$ARI = \frac{\sum_{kk'} \binom{n_{kk'}}{2} - [\sum_k \binom{a_k}{2} \sum_{k'} \binom{b_k}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_k \binom{a_k}{2} \sum_{k'} \binom{b_k}{2}] - [\sum_k \binom{a_k}{2} \sum_{k'} \binom{b_k}{2}]/\binom{n}{2}} \tag{3.21}$$

where $k = 1, \cdots, K$, and ARI can yield values between -1 and 1. The ARI is 1 when classifying perfectly, 0 means random labeling and a negative value indicates less than the expected random labelling. So, a positive value of ARI close to 1 indicates better classification accuracy.

Chapter 4

Simulation Studies

4.1 Simulation Design

First, we performed a simulation study to mimic the bacteria counts of a genus collected from gut microbiome samples, where we used two different sample sizes $n = (100, 300)$ and 100 repetitions. Since there are 113 observations in the study of Nakatsu et al. (2015), we selected a sample size range that contains 113. In order to mimic $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^T$ which is the health status in the real data, we simulated $\boldsymbol{x}$ from Bernoulli(0.5). Then, $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$ can be calculated by some specified values of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ (see details in Tables 4.1, 4.2 and 4.3). Next, Z_{ik}'s are simulated from the multinomial $(1, \boldsymbol{\pi})$ distribution resulting in an $n \times K$ matrix of z's with only 0 and 1 to represent the actual component for each subject. Given the membership $\boldsymbol{Z}$ vector and the Poisson mean parameters $\boldsymbol{\theta}$, we generated the count data $\boldsymbol{y}$. The R code for the simulation is in Appendix B.1. Simulations can be separated into three different component scenarios:

- Scenario 1: 2 components ZIP regression model;
- Scenario 2: 3 components GZIP regression mixture model;

- Scenario 3: 4 components GZIP regression mixture model.

For the 2 components ZIP regression model and the 3 components GZIP regression mixture model simulations, there are three groups of β and three groups of γ. Thus, with the cross combination, in total, we have 9 combinations in each scenario. For the 4 components GZIP model simulation, two groups of β and two groups of γ were used resulting in 4 combinations. There are a total of 12 paramaters, 6 β's and 6 γ's, respectively, in a 4 components model. A small change of one parameter influences the weights and means in the model with a small sample size of 100, thus the model fitting is very sensitive to the parameter settings. Therefore, we only consider 4 combinations of parameters values for the 4 components model. The parameter values used in simulations are provided in Table 4.1, Table 4.2 and Table 4.3. The parameter sets are from the analyses of real data.

	Combination 1				**Combination 2**				**Combination 3**			
Ture value	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
	-1.753	0.318	2.036	0.736	-1.753	0.318	2.029	0.170	-1.753	0.318	3.673	0.718
	Combination 4				**Combination 5**				**Combination 6**			
True value	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
	-2.215	1.011	2.029	0.170	-2.215	1.011	2.036	0.736	-2.215	1.011	3.673	0.718
	Combination 7				**Combination 8**				**Combination 9**			
True value	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
	-1.514	0.703	3.673	0.718	-1.514	0.703	2.036	0.736	-1.514	0.703	2.029	0.170

Table 4.1: *The parameter values used in 2 components ZIP regression mixture model simulation*

Combination 1								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	-0.252	0.609	-1.640	1.875	1.803	0.529	3.868	1.268
Combination 2								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	-0.252	0.609	-1.640	1.875	3.782	-1.102	5.131	-0.720
Combination 3								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	-0.252	0.609	-1.640	1.875	3.779	-0.820	4.533	-0.747
Combination 4								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.0467	0.359	-0.154	-0.133	3.782	-1.102	5.131	-0.720
Combination 5								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.0467	0.359	-0.154	-0.133	1.803	0.529	3.868	1.268
Combination 6								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.0467	0.359	-0.154	-0.133	3.780	-0.820	4.533	-0.747
Combination 7								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.904	-0.170	-0.514	0.737	3.779	-0.820	4.533	-0.747
Combination 8								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.904	-0.170	-0.514	0.737	1.803	0.529	3.868	1.268
Combination 9								
True value	β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
	0.904	-0.170	-0.514	0.737	3.782	-1.102	5.131	-0.720

Table 4.2: *The parameter values used in 3 components GZIP regression mixture model simulation*

Combination 1												
True value	β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
	-1.633	1.373	-1.360	0.352	-1.675	0.825	0.804	-0.102	2.527	0.418	3.711	0.766
Combination 2												
True value	β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
	-2.124	2.457	-1.875	1.023	-1.533	1.558	1.012	-0.253	3.245	0.721	3.771	1.249
Combination 3												
True value	β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
	-1.633	1.373	-1.360	0.352	-1.675	0.825	1.012	-0.253	3.245	0.721	3.771	1.249
Combination 4												
True value	β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
	-2.124	2.457	-1.875	1.023	-1.533	1.558	0.804	-0.102	2.527	0.418	3.711	0.766

Table 4.3: *The parameter values used in 4 components GZIP regression mixture model simulation*

To evaluate performance, the empirical mean, percent relative bias (PRB), empirical standard error (ESE) and mean squared error (MSE) for each estimated parameter based on the 100 repetitions are used. Empirical mean is the average of the estimated values and it should be close to the true value. The percent relative bias is the percentage bias from the true value. Empirical standard error is calculated as the square root of the variance of the estimates. Mean squared error measures the average of the square of the error. The calculation equations for these four metrics are:

- $Mean(\hat{\lambda}) = \frac{1}{r}\sum_{j=1}^{r}\hat{\lambda}_j$
- $PRB(\hat{\lambda}) = \frac{|\hat{\lambda}-\lambda|}{\lambda} \times 100\%$
- $ESE(\hat{\lambda}) = \sqrt{\hat{Var}(\hat{\lambda})} = \sqrt{\frac{1}{r-1}\sum_{j=1}^{r}(\hat{\lambda}_j - \bar{\lambda})^2}$
- $MSE(\hat{\lambda}) = \frac{1}{r}\sum_{j=1}^{r}(\hat{\lambda}_j - \lambda)^2$

where λ is the true parameter value, $\hat{\lambda}$ is the estimated parameter value, $\bar{\lambda}$ is the average of the parameter estimates and r is the number of repetitions.

In simulations performed by Lim et al. (2014), only the 3 components GZIP regression mixture model is considered with sample sizes of 300, 500 and 1000 which are much larger than our dataset. In our simulation study, we also evaluate the performance of the GZIP regression mixture model when the number of components K is smaller (e.g. $K = 2$) and larger (e.g. $K = 4$). At the same time, we consider the sample size of $n = 100$ which is similar to the real data.

4.2 Simulation Results

Table 4.4 presents the simulation results for the 2 components ZIP regression model. Tables 4.5 and 4.6 present simulation results for the 3 components and 4 components GZIP models, respectively.

From all tables, it is noted that when the sample size is 300, most of the empirical means are closer to true values than the situations when sample size is 100 because of the smaller percent relative bias. Likewise, the ESE and MSE are also smaller when the sample size is larger. We also find that the simulation results of 2 and 3 components models are better than the 4 components model when looking at the values of ESE and MSE.

		Combination 1				**Combination 2**				**Combination 3**			
True value		β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
		-1.753	0.318	2.036	0.736	-1.753	0.318	2.029	0.170	-1.753	0.318	3.673	0.718
$n = 100$	mean	-1.808	0.377	2.021	0.763	-1.795	0.356	2.036	0.168	-1.778	0.359	3.666	0.722
	PRB	-3.14%	18.55%	0.74%	3.67%	-2.40%	11.95%	0.34%	1.18%	-1.43%	12.89%	0.19%	0.56%
	ESE	0.048	0.057	0.018	0.019	0.040	0.062	0.016	0.021	0.041	0.050	0.007	0.007
	MSE	0.235	0.323	0.031	0.037	0.156	0.378	0.026	0.042	0.167	0.247	0.004	0.005
$n = 300$	mean	-1.792	0.317	2.035	0.750	-1.778	0.289	2.047	0.157	-1.736	0.291	3.667	0.723
	PRB	-2.22%	0.31%	0.05%	1.90%	-1.43%	9.12%	0.89%	7.65%	-0.97%	8.49%	0.16%	0.70%
	ESE	0.026	0.033	0.007	0.009	0.026	0.037	0.008	0.010	0.025	0.032	0.004	0.004
	MSE	0.069	0.107	0.005	0.009	0.070	0.138	0.007	0.011	0.062	0.099	0.001	0.002
		Combination 4				**Combination 5**				**Combination 6**			
True value		β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
		-2.215	1.011	2.029	0.170	-2.215	1.011	2.036	0.736	-2.215	1.011	3.673	0.718
$n = 100$	mean	-2.352	1.108	2.001	0.211	-2.329	1.124	2.039	0.724	-2.326	1.056	3.659	0.732
	PRB	-6.19%	9.59%	1.38%	24.12%	-5.15%	11.18%	0.15%	1.63%	-5.01%	4.45%	0.38%	1.95%
	ESE	0.058	0.064	0.020	0.022	0.051	0.060	0.019	0.021	0.048	0.062	0.008	0.009
	MSE	0.357	0.413	0.042	0.051	0.266	0.370	0.037	0.045	0.237	0.381	0.006	0.009
$n = 300$	mean	-2.262	1.051	2.023	0.187	-2.305	1.141	2.036	0.728	-2.286	1.055	3.674	0.720
	PRB	-2.12%	3.96%	0.30%	10%	-4.06%	12.86%	0%	1.09%	-3.21%	4.35%	0.27%	0.28%
	ESE	0.031	0.037	0.009	0.012	0.033	0.038	0.008	0.009	0.030	0.037	0.004	0.005
	MSE	0.099	0.135	0.009	0.014	0.119	0.163	0.007	0.008	0.092	0.137	0.002	0.002
		Combination 7				**Combination 8**				**Combination 9**			
True value		β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}	β_{01}	β_{11}	γ_{01}	γ_{11}
		-1.514	0.703	3.673	0.718	-1.514	0.703	2.036	0.736	-1.514	0.703	2.029	0.170
$n = 100$	mean	-1.547	0.726	3.683	0.706	-1.547	0.700	2.036	0.745	-1.587	0.762	2.028	0.163
	PRB	-2.18%	3.27%	0.27%	1.67%	-2.18%	0.43%	0%	1.22%	-4.82%	8.39%	0.05%	4.12%
	ESE	0.040	0.047	0.005	0.006	0.043	0.048	0.011	0.013	0.041	0.050	0.014	0.017
	MSE	0.157	0.218	0.003	0.003	0.187	0.232	0.012	0.017	0.176	0.246	0.020	0.028
$n = 300$	mean	-1.538	0.722	3.676	0.713	-1.536	0.712	2.035	0.741	-1.507	0.681	2.040	0.164
	PRB	-1.59%	2.70%	0.08%	0.70%	-1.45%	1.28%	0.05%	0.68%	-0.46%	3.13%	0.54%	3.53%
	ESE	0.022	0.026	0.003	0.004	0.020	0.024	0.006	0.007	0.021	0.026	0.008	0.010
	MSE	0.047	0.069	0.001	0.002	0.040	0.058	0.004	0.005	0.044	0.067	0.006	0.010

Table 4.4: *Simulation results of mean, PRB (percent relative bias), ESE (empirical standard error) and MSE (mean squared error) for 2 components ZIP regression mixture model*

Combination 1									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		-0.252	0.609	-1.640	1.875	1.803	0.529	3.868	1.268
$n = 100$	mean	-0.309	0.684	-1.620	1.860	1.806	0.531	3.863	1.272
	PRB	-22.62%	12.32%	-1.22%	0.80%	0.17%	0.38%	0.13%	0.32%
	ESE	0.033	0.049	0.059	0.067	0.009	0.012	0.007	0.008
	MSE	0.109	0.244	0.345	0.439	0.007	0.014	0.005	0.006
$n = 300$	mean	-0.267	0.570	-1.655	1.844	1.799	0.529	3.866	1.269
	PRB	-5.95%	6.40%	-0.92%	1.65%	0.22%	0%	0.13%	0.08%
	ESE	0.020	0.030	0.033	0.041	0.006	0.008	0.004	0.004
	MSE	0.040	0.093	0.106	0.168	0.004	0.006	0.002	0.002
Combination 2									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		-0.252	0.609	-1.640	1.875	3.782	-1.102	5.131	-0.720
$n = 100$	mean	-0.287	0.672	-1.670	1.928	3.781	-1.092	5.124	-0.713
	PRB	-13.89%	10.34%	-1.83%	2.83%	0.03%	-0.91%	0.14%	-0.97%
	ESE	0.034	0.052	0.052	0.062	0.004	0.008	0.003	0.004
	MSE	0.118	0.274	0.270	0.382	0.001	0.006	0.001	0.002
$n = 300$	mean	-0.268	0.636	-1.663	1.886	3.780	-1.096	5.132	-0.720
	PRB	-6.35%	4.43%	-1.40%	0.59%	0.05%	-0.54%	0.02%	0%
	ESE	0.019	0.028	0.031	0.040	0.002	0.004	0.002	0.002
	MSE	0.035	0.079	0.093	0.157	0.0004	0.001	0.0004	0.001
Combination 3									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		-0.252	0.609	-1.640	1.875	3.779	-0.820	4.533	-0.747
$n = 100$	mean	-0.254	0.614	-1.676	1.916	3.777	-0.819	4.535	-0.750
	PRB	-0.79%	0.82%	-2.20%	2.19%	0.05%	-0.12%	0.04%	-0.40%
	ESE	0.029	0.044	0.050	0.059	0.003	0.007	0.005	0.006
	MSE	0.081	0.193	0.253	0.344	0.001	0.005	0.002	0.004
$n = 300$	mean	-0.274	0.595	-1.684	1.880	3.777	-0.820	4.525	-0.736
	PRB	-8.73%	2.30%	-2.68%	0.27%	0.05%	0%	0.18%	-1.47%
	ESE	0.018	0.027	0.030	0.036	0.002	0.004	0.003	0.003
	MSE	0.033	0.073	0.090	0.129	0.0003	0.001	0.001	0.001

Table 4.5: *Simulation results of mean, PRB (percent relative bias), ESE (empirical standard error) and MSE (mean squared error) for 3 components GZIP regression mixture model (continued)*

Combination 4									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.0467	0.359	-0.154	-0.133	3.782	-1.102	5.131	-0.720
$n = 100$	mean	0.039	0.375	-0.209	-0.093	3.784	-1.106	5.131	-0.717
	PRB	16.49%	4.46%	-35.71%	-30.06%	0.05%	-0.36%	0%	-0.42%
	ESE	0.035	0.043	0.037	0.049	0.004	0.007	0.002	0.003
	MSE	0.119	0.186	0.142	0.242	0.002	0.005	0.0005	0.001
$n = 300$	mean	0.051	0.362	-0.186	-0.120	3.781	-1.099	5.130	-0.721
	PRB	9.21%	0.84%	-20.78%	-9.77%	0.03%	-0.27%	0.02%	-0.14%
	ESE	0.019	0.025	0.019	0.027	0.002	0.003	0.001	0.002
	MSE	0.034	0.062	0.038	0.072	0.0005	0.001	0.0001	0.0004
Combination 5									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.0467	0.359	-0.154	-0.133	1.803	0.529	3.868	1.268
$n = 100$	mean	0.037	0.377	-0.171	-0.071	1.804	0.539	3.873	1.263
	PRB	20.77%	5.01%	-11.04%	-46.62%	0.06%	1.89%	0.13%	0.39%
	ESE	0.039	0.052	0.034	0.053	0.010	0.011	0.004	0.005
	MSE	0.151	0.266	0.113	0.283	0.011	0.012	0.002	0.002
$n = 300$	mean	0.047	0.360	-0.154	-0.150	1.796	0.528	3.865	1.270
	PRB	0.64%	0.28%	0%	-12.78%	0.39%	0.19%	0.08%	0.16%
	ESE	0.017	0.024	0.019	0.033	0.006	0.007	0.002	0.003
	MSE	0.030	0.056	0.037	0.108	0.003	0.005	0.0005	0.001
Combination 6									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.0467	0.359	-0.154	-0.133	3.780	-0.820	4.533	-0.747
$n = 100$	mean	0.049	0.390	-0.143	-0.101	3.780	-0.825	4.527	-0.746
	PRB	4.93%	8.64%	-7.14%	-24.06%	0%	-0.61%	0.13%	-0.13%
	ESE	0.040	0.053	0.041	0.059	0.004	0.006	0.003	0.006
	MSE	0.157	0.283	0.168	0.349	0.002	0.004	0.001	0.003
$n = 300$	mean	0.045	0.360	-0.147	-0.145	3.782	-0.827	4.535	-0.748
	PRB	3.64%	0.28%	-4.55%	-9.02%	0.05%	-0.85%	0.04%	-0.13%
	ESE	0.022	0.030	0.020	0.032	0.002	0.004	0.002	0.003
	MSE	0.050	0.088	0.041	0.102	0.0004	0.001	0.0003	0.001

Table 4.5: *Simulation results of mean, PRB (percent relative bias), ESE (empirical standard error) and MSE (mean squared error) for 3 components GZIP regression mixture model (continued)*

Combination 7									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.904	-0.170	-0.514	0.737	3.779	-0.820	4.533	-0.747
$n = 100$	mean	0.914	-0.144	-0.509	0.771	3.776	-0.815	4.513	-0.724
	PRB	1.11%	-15.29%	-0.97%	4.61%	0.08%	-0.61%	0.44%	-3.08%
	ESE	0.037	0.052	0.055	0.069	0.003	0.005	0.008	0.009
	MSE	0.133	0.271	0.295	0.469	0.001	0.003	0.006	0.009
$n = 300$	mean	0.909	-0.187	-0.551	0.761	3.776	-0.825	4.533	-0.749
	PRB	0.55%	-10%	-7.20%	3.26%	0.08%	-0.61%	0%	-0.27%
	ESE	0.019	0.027	0.029	0.039	0.002	0.003	0.002	0.003
	MSE	0.034	0.073	0.085	0.150	0.0003	0.001	0.0005	0.001
Combination 8									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.904	-0.170	-0.514	0.737	1.803	0.529	3.868	1.268
$n = 100$	mean	0.954	-0.194	-0.560	0.824	1.800	0.526	3.865	1.265
	PRB	5.53%	-14.12%	-8.95%	11.80%	0.17%	0.57%	0.08%	0.24%
	ESE	0.033	0.048	0.048	0.059	0.007	0.010	0.006	0.007
	MSE	0.107	0.228	0.229	0.355	0.005	0.011	0.004	0.004
$n = 300$	mean	0.913	-0.169	-0.525	0.740	1.805	0.521	3.866	1.267
	PRB	1.00%	-0.59%	-2.14%	0.41%	0.11%	1.51%	0.05%	0.08%
	ESE	0.019	0.027	0.029	0.035	0.004	0.006	0.003	0.003
	MSE	0.036	0.072	0.081	0.118	0.002	0.003	0.001	0.001
Combination 9									
True value		β_{01}	β_{11}	β_{02}	β_{12}	γ_{01}	γ_{11}	γ_{02}	γ_{12}
		0.904	-0.170	-0.514	0.737	3.782	-1.102	5.131	-0.720
$n = 100$	mean	0.947	-0.191	-0.546	0.725	3.785	-1.111	5.117	-0.703
	PRB	4.76%	-12.35%	-6.23%	1.63%	0.08%	-0.82%	0.27%	-2.36%
	ESE	0.037	0.051	0.053	0.066	0.003	0.006	0.013	0.013
	MSE	0.134	0.261	0.276	0.431	0.001	0.003	0.016	0.016
$n = 300$	mean	0.927	-0.168	-0.527	0.746	3.780	-1.099	5.130	-0.717
	PRB	2.54%	-1.18%	-2.53%	1.22%	0.05%	-0.27%	0.02%	-0.42%
	ESE	0.021	0.031	0.027	0.038	0.002	0.003	0.002	0.002
	MSE	0.045	0.094	0.070	0.140	0.0003	0.001	0.0002	0.001

Table 4.5: *Simulation results of mean, PRB (percent relative bias), ESE (empirical standard error) and MSE (mean squared error) for 3 components GZIP regression mixture model*

Combination 1													
True value		β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
		-1.633	1.373	-1.360	0.352	-1.675	0.825	0.804	-0.102	2.527	0.418	3.711	0.766
$n = 100$	mean	-1.563	1.285	-1.361	0.298	-1.671	0.772	0.657	0.018	2.533	0.400	3.716	0.762
	PRB	-4.29%	6.41%	-0.07%	15.34%	-0.24%	6.42%	18.28%	-117.65%	0.24%	4.31%	0.13%	0.52%
	ESE	0.068	0.090	0.045	0.068	0.051	0.067	0.066	0.072	0.015	0.019	0.007	0.008
	MSE	0.462	0.802	0.199	0.467	0.260	0.453	0.452	0.525	0.021	0.035	0.006	0.007
$n = 300$	mean	-1.640	1.452	-1.360	0.376	-1.705	0.850	0.726	-0.060	2.527	0.417	3.707	0.766
	PRB	-0.43%	5.75%	0%	6.82%	-1.79%	3.03%	9.70%	-41.18%	0%	0.24%	0.11%	0%
	ESE	0.028	0.036	0.024	0.036	0.034	0.042	0.031	0.034	0.007	0.008	0.004	0.004
	MSE	0.080	0.133	0.057	0.128	0.116	0.175	0.104	0.113	0.004	0.007	0.001	0.002
Combination 2													
True value		β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
		-2.124	2.457	-1.875	1.023	-1.533	1.558	1.012	-0.253	3.245	0.721	3.771	1.249
$n = 100$	mean	-2.028	2.436	-1.918	1.069	-1.625	1.714	0.825	-0.102	3.240	0.717	3.766	1.253
	PRB	-4.52%	0.85%	-2.29%	4.50%	-6.00%	10.01%	18.48%	59.68%	0.15%	0.55%	0.13%	0.32%
	ESE	0.081	0.097	0.065	0.090	0.053	0.076	0.074	0.080	0.012	0.014	0.008	0.009
	MSE	0.652	0.929	0.419	0.798	0.290	0.603	0.576	0.660	0.014	0.019	0.007	0.008
$n = 300$	mean	-2.151	2.489	-1.959	1.107	-1.583	1.682	0.996	-0.259	3.235	0.735	3.767	1.255
	PRB	-1.27%	1.30%	-4.48%	8.21%	-3.26%	7.96%	1.58%	-2.37%	0.31%	1.94%	0.11%	0.48%
	ESE	0.034	0.040	0.036	0.047	0.027	0.033	0.019	0.022	0.007	0.008	0.004	0.004
	MSE	0.117	0.157	0.135	0.229	0.075	0.126	0.035	0.049	0.005	0.006	0.002	0.002
Combination 3													
True value		β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
		-1.633	1.373	-1.360	0.352	-1.675	0.825	1.012	-0.253	3.245	0.721	3.771	1.249
$n = 100$	mean	-1.633	1.445	-1.477	0.485	-1.784	0.999	0.963	-0.281	3.230	0.734	3.767	1.252
	PRB	0%	5.24%	-8.60%	37.78%	-6.51%	21.09%	4.84%	-11.07%	0.46%	1.80%	0.11%	0.24%
	ESE	0.057	0.072	0.056	0.080	0.053	0.067	0.047	0.054	0.009	0.012	0.012	0.012
	MSE	0.317	0.512	0.322	0.656	0.287	0.473	0.218	0.294	0.009	0.014	0.014	0.014
$n = 300$	mean	-1.633	1.363	-1.343	0.355	-1.742	0.925	0.997	-0.226	3.246	0.719	3.774	1.243
	PRB	0%	0.73%	-1.25%	0.85%	-4.00%	12.12%	1.48%	-10.67%	0.03%	0.28%	0.08%	0.48%
	ESE	0.026	0.033	0.025	0.034	0.028	0.033	0.017	0.021	0.005	0.005	0.005	0.005
	MSE	0.065	0.106	0.064	0.114	0.083	0.117	0.029	0.045	0.002	0.003	0.003	0.003
Combination 4													
True value		β_{01}	β_{11}	β_{02}	β_{12}	β_{03}	β_{13}	γ_{01}	γ_{11}	γ_{02}	γ_{12}	γ_{03}	γ_{13}
		-2.124	2.457	-1.875	1.023	-1.533	1.558	0.804	-0.102	2.527	0.418	3.711	0.766
$n = 100$	mean	-1.960	2.358	-2.040	1.130	-1.511	1.529	0.774	-0.048	2.531	0.398	3.709	0.773
	PRB	-7.72%	4.03%	-8.80%	10.46%	-1.44%	1.86%	3.73%	52.94%	0.16%	4.78%	0.05%	0.91%
	ESE	0.063	0.082	0.068	0.096	0.046	0.063	0.074	0.077	0.026	0.028	0.006	0.007
	MSE	0.421	0.680	0.487	0.932	0.206	0.394	0.544	0.597	0.065	0.078	0.004	0.005
$n = 300$	mean	-2.148	2.502	-1.902	1.101	-1.545	1.593	0.748	-0.043	2.540	0.399	3.706	0.771
	PRB	-1.13%	1.83%	-1.44%	7.62%	-0.78%	2.25%	6.97%	-57.84%	0.51%	4.55%	0.13%	0.65%
	ESE	0.038	0.049	0.026	0.047	0.024	0.033	0.031	0.033	0.009	0.010	0.004	0.004
	MSE	0.141	0.245	0.069	0.225	0.059	0.112	0.101	0.112	0.009	0.011	0.001	0.002

Table 4.6: *Simulation results of mean, PRB (percent relative bias), ESE (empirical standard error) and MSE (mean squared error) for 4 components GZIP regression mixture model*

4.3 Performance of the Model Selection

As explored by Lim et al. (2014), we conducted simulations to investigate the performance of the GZIP regression mixture model selection. The first simulation is to check whether the BIC can identify the correct model. We use all of the parameter pairs as in Section 4.2 to generate data of size $n = 100$ from 2, 3, and 4 components models. Then, we fit each set of data with the 2 components ZIP model and the 3 and 4 components GZIP models where this is replicated 100 times. Table 4.7 is the result of the simulation for the 2 components ZIP model, Table 4.8 is for the 3 components GZIP models and Table 4.9 is for the 4 components GZIP model. These tables show that the BIC does very well for selecting the number of model components. All of the accuracies are 100%. Thus, the BIC is a valuable criterion for component identification.

True model	Fitted model			Accuracy rate
$k = 2$	$k = 2$	$k = 3$	$k = 4$	(%)
Combination 1	100	0	0	100
Combination 2	100	0	0	100
Combination 3	100	0	0	100
Combination 4	100	0	0	100
Combination 5	100	0	0	100
Combination 6	100	0	0	100
Combination 7	100	0	0	100
Combination 8	100	0	0	100
Combination 9	100	0	0	100

Table 4.7: *2 components GZIP Model selection simulation results based on BIC*

Secondly, the ARI is used to evaluate the classification performance of the models. We use each of the combinations from the different components models and compare the number of subjects classified into different components with their true positions. Then, we get 100 misclassification tables for each combination. After calculating the ARI for each table, the average number can represent the misclassification performance for each combination.

Table 4.10 are examples of the misclassification tables for the 2 components model for

True model $k=3$	Fitted model $k=2$	$k=3$	$k=4$	accuracy rate (%)
Combination 1	0	100	0	100
Combination 2	0	100	0	100
Combination 3	0	100	0	100
Combination 4	0	100	0	100
Combination 5	0	100	0	100
Combination 6	0	100	0	100
Combination 7	0	100	0	100
Combination 8	0	100	0	100
Combination 9	0	100	0	100

Table 4.8: *3 components GZIP Model selection simulation results based on BIC*

True model $k=4$	Fitted model $k=2$	$k=3$	$k=4$	accuracy rate (%)
Combination 1	0	0	100	100
Combination 2	0	0	100	100
Combination 3	0	0	100	100
Combination 4	0	0	100	100

Table 4.9: *4 components GZIP Model selection simulation results based on BIC*

Combination 1 such that $\beta = (-1.753, 0.318)^T$ and $\gamma = (2.036, 0.736)^T$. The corresponding weights are $(\pi_1, \pi_2) = (0.2,\ 0.8)$. In the simulation results, since the ARI values for the 2 components model are all 1, no subject is misclassified into incorrect components.

Real Component	**Classified**	
	1	2
1	15	0
2	0	85

Real Component	**Classified**	
	1	2
1	20	0
2	0	80

Real Component	**Classified**	
	1	2
1	24	0
2	0	76

Table 4.10: *Examples of 2 components model misclassification tables*

For the 3 components model, Table 4.11 provides examples of misclassification tables for the 3 components model with the parameter pair $\beta = (-0.252, 0.609, -1.640, 1.875)^T$ and $\gamma = (1.803, 0.529, 3.868, 1.268)^T$. The corresponding weights are $(\pi_1, \pi_2, \pi_3) = (0.35,\ 0.25,\ 0.40)$. In this combination, two cases are misclassified and ARI values for these two results are both 0.965. Thus, the average ARI for the 3 components model is 0.9993 which is quite

close to 1. Table 4.12 presents results for all 3 components model combinations.

Real Component	Classified		
	1	2	3
1	36	0	0
2	0	26	0
3	0	0	38

Real Component	Classified		
	1	2	3
1	36	0	**1**
2	0	18	0
3	0	0	45

Real Component	Classified		
	1	2	3
1	31	0	**1**
2	0	32	0
3	0	0	36

Table 4.11: *Examples of 3 components model misclassification tables*

Combination	Adjusted Rand Index (ARI)
1	0.9993
2	1
3	0.9889
4	1
5	0.9981
6	0.9894
7	0.9890
8	0.9990
9	1

Table 4.12: *ARI results of 3 components model misclassification table*

In Table 4.13, there are three examples of misclassification tables for the 4 components model with the parameter pair in Combination 1, which is $\beta = (-1.633, 1.373, -1.360, 0.352, -1.675, 0.825)$ and $\gamma = (0.804, -0.102, 2.527, 0.418, 3.711, 0.766)$. The corresponding weights are $(\pi_1, \pi_2, \pi_3, \pi_4) = (0.20, 0.15, 0.15, 0.50)$. Table 4.14 presents results of the 4 components GZIP model combinations. We find that the ARI values for the 4 components model are lower than the ARI values for the 2 and 3 components models. This is because the sample size is only 100 and for more components, it is difficult to correctly classify subjects when two of the components are similar.

Real Component	Classified			
	1	2	3	4
1	20	0	0	0
2	0	15	0	0
3	0	0	16	0
4	0	0	0	49

Real Component	Classified			
	1	2	3	4
1	19	**1**	0	0
2	0	12	0	0
3	0	0	14	0
4	0	0	0	54

Real Component	Classified			
	1	2	3	4
1	14	0	0	**6**
2	**1**	15	0	0
3	0	0	15	0
4	0	0	0	49

Table 4.13: *Examples of 4 components model misclassification tables*

Combination	Adjusted Rand Index (ARI)
1	0.9244
2	0.9127
3	0.9384
4	0.9274

Table 4.14: *ARI results of 4 components model misclassification table*

Chapter 5

Real Data Introduction and Results

5.1 Application to Gut Microbiome Data

In the development of colorectal cancer (CRC), gut mucosal microbiome has a significant influence (Nakatsu et al., 2015). This application uses the dataset retrieved from the Prince of Wales Hospital of the Chinese University of Hong Kong and the First Affiliated Hospital of the Sun Yat-Sen University (Nakatsu et al., 2015).

The gut mucosal microbiome data includes 113 subjects comprising 61 healthy controls and 52 carcinoma samples. The number of bacterial genera we consider in this thesis is 86. The dataset catalogues the microbial communities in gut mucosa into a phylogenetic tree for the hierachical order of taxonomic rank as Domain, Kingdom, Phylum, Class, Order, Family, Genus and Species. In this research, we only consider the genus taxa. The feature of this dataset is that all observations are count data. Most of them have many zeros but some taxa have large non-zero counts and few zeros. A subset of the gut mucosal microbiome data is in Appendix A.

After fitting 2, 3, and 4 components GZIP regression mixture models to each genus of the gut microbiome data and selecting the model with the lowest BIC, the table of results can

be found in Table 5.1. The models considered could not be fit to 37 bacterial genera. For the remaining genera, the 2 components model fits 3 genera best, the 3 components model fits 8 genera best, and the 4 components model fits 38 genera best. In 86 genera, some referred to as unclassified are those where there is no information at the genus level.

Component Number	**Bacterial gene taxa**
2	Odoribacter, Pedobacter, Gemella
3	Eggerthella, Staphylococcus, Enterococcus, Christensenellaceae˙R-7˙group, Aeromonas, Coriobacteriaceae/unclassified, Bacteroidetes_unclassified/unclassified, Bacillales_unclassified/unclassified
4	Paraprevotella, Lactobacillus, Anaerococcus, Peptoniphilus Mogibacterium, [Eubacterium]_hallii_group, Coprococcus_1, Coprococcus_3, Fusicatenibacter, Lachnospiraceae_NK4A136_group, Pseudobutyrivibrio, Roseburia, Tyzzerella, Peptostreptococcus, Romboutsia, [Eubacterium]_coprostanoligenes_group, Pseudoflavonifractor, Ruminococcaceae_UCG-002, Ruminococcaceae_UCG-005, Ruminococcaceae_UCG-008, Ruminococcaceae_UCG-014, Subdoligranulum, Dialister, Parasutterella, Neisseria, Campylobacter, Pseudomonas, Lachnoclostridium, Actinobacteria_unclassified/unclassified, Bacteroidales_unclassified/unclassified, Clostridiales_unclassified/unclassified, Lachnospiraceae/unclassified, Peptostreptococcaceae/unclassified, Fusobacteriaceae/unclassified, Fusobacteriales˙unclassified/unclassified, Pasteurellaceae/unclassified, Pseudomonadaceae/unclassified, Proteobacteria_unclassified/unclassified

Table 5.1: *The model selection results of gut microbiome data*

5.2 Special Cases

Since the models could not be fitted to 37 bacteria genera, we investigated histograms of their OTU counts. Two sets of examples are given in Figures 5.1 and 5.2 to represent two special cases. One special case is that almost all of the count data are zero so there is insufficient information to fit the other Poisson components of the model. The other is that there are few zeros, and thus, the OTU data is no longer zero-inflated. In this case, a mixture of Poisson regression models should be used instead.

Figure 5.1: *Special case 1: few non-zero counts*

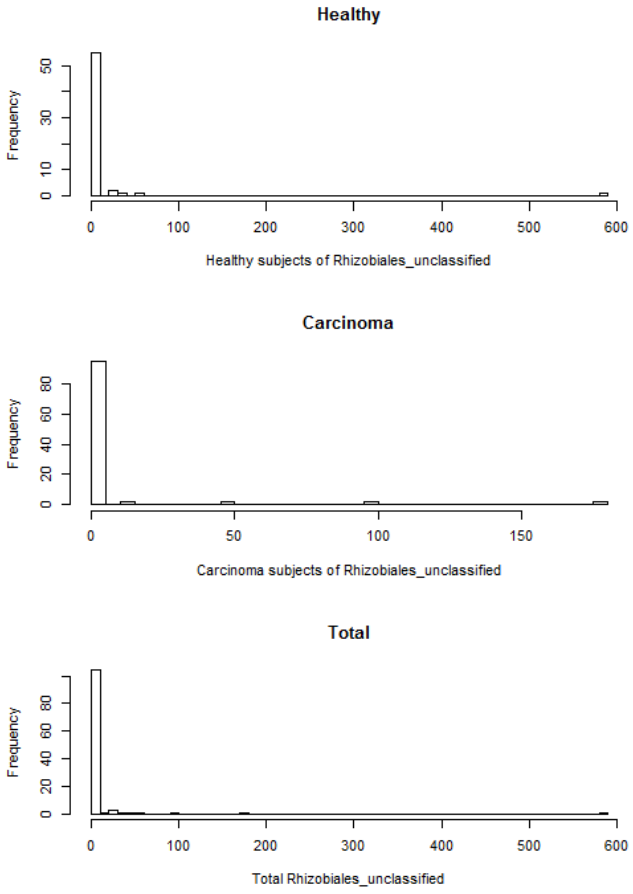
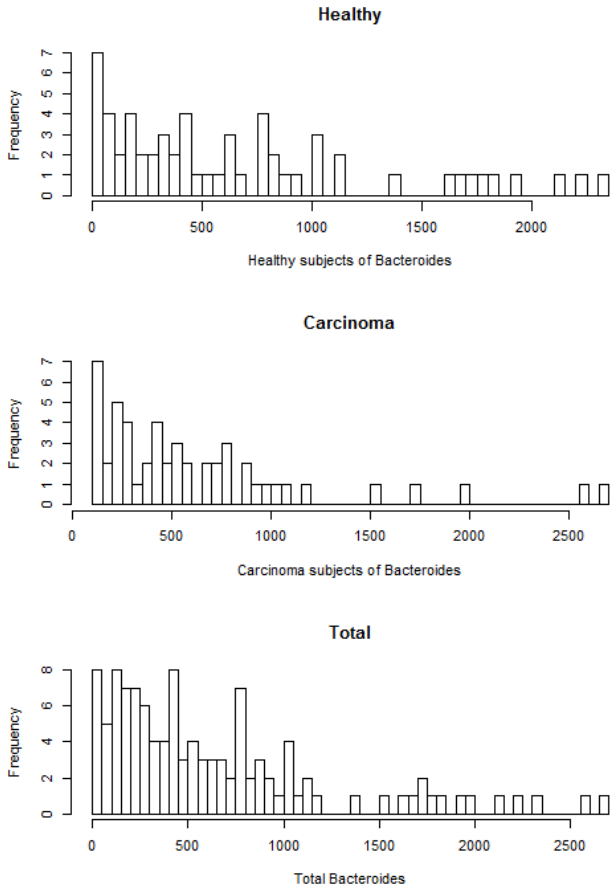


Figure 5.2: *Special case 2: few zeros*



Chapter 6

Conclusion and Future Work

We have used GZIP regression mixture models to fit the zero-inflated count data. The EM algorithm is chosen to estimate the unknown parameters and the BIC is used to select the number of components. Our simulation studies show that this model is appropriate for estimating parameters and that using the BIC to choose the number of components works extremely well. In addition, with increasing sample size, the results are improved with smaller ESE and MSE. The models are applied to gut mucosal microbiome communities for two groups of subjects, carcinoma patients and healthy individuals, from the study of Nakatsu et al. (2015). In this dataset, we only consider the disease status as a covariate. The 2 components model fits 3 genera best, the 3 components model fits 8 genera best, and the 4 components model fits 38 genera best. No models fit the remaining 37 genera well.

For these genera that can be fitted well by the GZIP regression mixture model, an association test for the association between the disease status and the bacteria can be established based on it. In this case, we can do an overall test as:

$$H_0 : \boldsymbol{\beta}_{-0} = 0, \boldsymbol{\gamma}_{-0} = 0 \quad vs \quad H_a : \boldsymbol{\beta}_{-0} \neq 0 \quad or/and \quad \boldsymbol{\gamma}_{-0} \neq 0.$$

For future work, we can first consider the genera listed in Table 5.1. Within each Poisson component, we assume the variance and mean are equal. However, this assumption may not be valid. In order to address this component-specific overdispersion problem, the generalized zero-inflated negative binomial regression mixture model can be considered (Rodrıguez, 2013). Simulation studies and performance assessment can also be done to compare it with the current proposed GZIP regression mixture models.

Secondly, a better model for fitting the special cases should also be developed. As already mentioned in Section 5.2, there are two kinds of special cases. One is when almost all counts are zeros. The other is when there are very few zero counts. We can focus on the latter circumstance where a reasonable assumption is that the zeros come from other Poisson distributions. Thus, we can assume that the data constitutes multiple Poisson distributions. The number of Poisson distributions can be any positive integer.

Thirdly, the Newton-Raphson method which is discussed in Section 3.2.1 can be an optional method compared with the EM algorithm. Since the reason we do not use the Newton-Raphson method is that it is computationally intensive, this method is still usable. In future work, a comparison of the EM algorithm and the Newton-Raphson method can be investigated.

Bibliography

Bäckhed, F., R. E. Ley, J. L. Sonnenburg, D. A. Peterson, and J. I. Gordon (2005). Host-bacterial mutualism in the human intestine. *Science 307*(5717), 1915–1920.

Bak, S. (2017). Generalized Linear Regression Model With LASSO And Group LASSO Regularization Methods For Predicting Disease Status Using The Microbiome Data. unpublished thesis from University of Guelph.

Burrus, C. S. (2012). Iterative reweighted least squares. *OpenStax-CNX Web site. http://cnx.org/content/m45285/1.12*.

Bybee, S. M., H. Bracken-Grissom, B. D. Haynes, R. A. Hermansen, R. L. Byers, M. J. Clement, J. A. Udall, E. R. Wilcox, and K. A. Crandall (2011). Targeted amplicon sequencing (tas): a scalable next-gen approach to multilocus, multitaxa phylogenetics. *Genome Biology And Evolution 3*, 1312–1323.

Carding, S., K. Verbeke, D. T. Vipond, B. M. Corfe, and L. J. Owen (2015). Dysbiosis of the gut microbiota in disease. *Microbial Ecology in Health and Disease 26*(1), 26191.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal Of The Royal Statistical Society. Series B (Methodological) 39*, 1–38.

Doctrow, B. (2016). Profile of Joachim Messing. *Proceedings of the National Academy of Sciences 113*(29), 7935–7937.

Gardner, R. C., A. J. Howarth, P. Hahn, M. Brown-Luedi, R. J. Shepherd, and J. Messing (1981). The complete nucleotide sequence of an infectious clone of cauliflower mosaic virus by m13mp7 shotgun sequencing. *Nucleic Acids Research 9*(12), 2871–2888.

Gregory, S. G. (2005). Contig Assembly. `http://www.els.net`. [doi: 10.1038/npg.els.0005365].

Grice, E. A., H. H. Kong, G. Renaud, A. C. Young, G. G. Bouffard, R. W. Blakesley, T. G. Wolfsberg, M. L. Turner, and J. A. Segre (2008). A diversity profile of the human skin microbiota. *Genome Research 18*(7), 1043–1050.

Janssen, P. H. (2006). Identifying the dominant soil bacterial taxa in libraries of 16s rrna and 16s rrna genes. *Applied and Environmental Microbiology 72*(3), 1719–1728.

Kumar, P. S., M. R. Brooker, S. E. Dowd, and T. Camerlengo (2011). Target region selection is a critical determinant of community fingerprints generated by 16s pyrosequencing. *PloS One 6*(6), e20956.

Lambert, D. (1992). Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics 34*(1), 1–14.

Ley, R. E., D. A. Peterson, and J. I. Gordon (2006). Ecological and evolutionary forces shaping microbial diversity in the human intestine. *Cell 124*(4), 837–848.

Lim, H. K., W. K. Li, and L. Philip (2014). Zero-inflated Poisson regression mixture model. *Computational Statistics & Data Analysis 71*, 151–158.

Maechler, M., P. Rousseeuw, A. Struyf, M. Hubert, and K. Hornik (2017). *Cluster: Cluster Analysis Basics and Extensions*. R package version 2.0.6 — For new features, see the 'Changelog' file (in the package source).

McLachlan, G. and T. Krishnan (2007). *The EM Algorithm and Extensions*, Volume 382. John Wiley & Sons.

McLachlan, G. and D. Peel (2004). *Finite Mixture Models*. John Wiley & Sons.

Moos, W. H., D. V. Faller, D. N. Harpp, I. Kanara, J. Pernokas, W. R. Powers, and K. Steliou (2016). Microbiota and neurological disorders: a gut feeling. *BioResearch Open Access 5*(1), 137–145.

Nakatsu, G., X. Li, H. Zhou, J. Sheng, S. H. Wong, W. K. K. Wu, S. C. Ng, H. Tsoi, Y. Dong, N. Zhang, et al. (2015). Gut mucosal microbiome across stages of colorectal carcinogenesis. *Nature Communications 6*, 8727.

R Core Team (2017). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association 66*(336), 846–850.

Rodrıguez, G. (2013). Models for count data with overdispersion. `http://data.princeton.edu/wws509/notes/c4a.pdf`. Retrieved from Princeton University.

Roeder, K. and L. Wasserman (1997). Practical Bayesian density estimation using mixtures of normals. *Journal of the American Statistical Association 92*(439), 894–902.

Sartor, R. B. and G. D. Wu (2017). Roles for intestinal bacteria, viruses, and fungi in pathogenesis of inflammatory bowel diseases and therapeutic approaches. *Gastroenterology 152*(2), 327–339.

Schloss, P. D., S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, et al. (2009). Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Applied and Environmental Microbiology 75*(23), 7537–7541.

Schwarz, G. et al. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*(2), 461–464.

Tamboli, C., C. Neut, P. Desreumaux, and J. Colombel (2004). Dysbiosis in inflammatory bowel disease. *Gut 53*(1), 1–4.

Turnbaugh, P. J., R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon (2007). The human microbiome project. *Nature 449*(7164), 804.

Ursell, L. K., J. L. Metcalf, L. W. Parfrey, and R. Knight (2012). Defining the human microbiome. *Nutrition Reviews 70*(suppl_1), S38–S44.

Venables, W. N. and B. D. Ripley (2002). *Modern Applied Statistics with S* (Fourth ed.). New York: Springer.

Yarza, P., P. Yilmaz, E. Pruesse, F. O. Glöckner, W. Ludwig, K.-H. Schleifer, W. B. Whitman, J. Euzéby, R. Amann, and R. Rosselló-Móra (2014). Uniting the classification of cultured and uncultured bacteria and archaea using 16s rrna gene sequences. *Nature Reviews Microbiology 12*(9), 635.

Appendix A

A Subset of the Gut Mucosal Microbiome Data

Carcinoma	Actinomyces	Collinsella	Eggerthella	unclassified	Odoribacter	$\cdots$
No	0	0	0	0	0	
No	5	22	2	1	0	
Yes	0	16	0	0	1	
No	3	23	0	1	0	
No	5	0	0	0	0	
No	7	260	0	1	19	
No	2	0	0	6	0	
No	0	0	0	212	0	
No	38	151	1	40	0	
No	0	0	0	15	0	
Yes	130	0	2	757	3	
No	0	0	3	33	0	
Yes	54	23	1	45	7	
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\cdots$
Yes	0	11	3	2	0	
Yes	1	0	18	0	6	
No	5	253	0	0	0	
No	0	0	0	0	0	
Yes	0	12	4	0	0	
No	0	0	0	3	0	
No	27	79	1	16	6	
No	3	139	0	18	15	
No	2	88	0	0	0	
No	0	145	0	9	0	
No	0	29	51	3	0	

Appendix B

R Source Code: 3 Components Model

B.1 Simulation

```
library(LaplacesDemon)
library(nnet)
# simulate data
# use function "multinom" in package "nnet"
dslnex=function(x,b01,b11,b02,b12,r01,r11,r02,r12,n){
  x=rbern(n,0.5)
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p3=1/(1+exp(b01+b11*x)+exp(b02+b12*x))
  pi=cbind(p1,p2,p3)
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  t3=rep(0,n)
  t=cbind(t1,t2,t3)
  z=t(apply(pi,1,function(w) rmultinom(1,1,w)))
  T=c()
  y=c()
```

```
  for (i in 1:n){
    T=rbind(T,c(rpois(1,t[i,1]),rpois(1,t[i,2]),0))
    y=rbind(y,z[i,]%*%T[i,])
  }
  count=cbind(x,y)
  return(count)
}

# EM algorithm:
# E step:
ntest=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
  # p and t are vectors
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p3=1 p1 p2
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  ztest=matrix(ncol=3,nrow=length(x))
  for (i in 1:length(x)){
    if (y[i]==0){
      z1=(p1[i]*exp( t1[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
      z2=(p2[i]*exp( t2[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
      ztest[i,]=c(z1,z2,1 z1 z2)
    } else if (y[i]>0) {
      z1=(p1[i]*dpois(y[i],t1[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i]))
      z2=(p2[i]*dpois(y[i],t2[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i]))
      ztest[i,]=c(z1,z2,0)
    }
  }
```

```
  return(ztest)
}

# likelihood function:
lic=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
  l=c()
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p3=1 p1 p2
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  for (i in 1:length(x)){
    if (y[i]>0) {
      l[i]=log(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],t2[i]))
    } else {
      l[i]=log(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i] )
    }
  }
  L=sum(l)
  c(L)
}

#stop criteria function:
AAC=function(L){
  if (length(L)<4){
    val=1
    return(val)
  } else {
    # b is the length of L
    b=length(L)
    Ck_1=(L[b] L[b 1])/(L[b 1] L[b 2])
```

```
    la_k1=L[b 1]+(L[b] L[b 1]) /(1 Ck_1)
    Ck=(L[b 1] L[b 2]) /(L[b 2] L[b 3])
    la_k=L[b 2]+(L[b 1] L[b 2]) /(1 Ck)
    val=abs(la_k1 la_k)
    if(is.nan(val)) val=0
    return(val)
  }
}

outcome=c()
for (i in 1:100){
  b1 =0.04668917
  b2 =0.35877746
  b3 = 0.15435475
  b4 = 0.13332753
  r1 =3.7794206
  r2 = 0.8203640
  r3 = 4.532756
  r4 = 0.7466757
  n=100
  simu=dslnex(x,b1,b2,b3,b4,r1,r2,r3,r4,n)
  x=simu[,1]
  y=simu[,2]
  #initial value:
  b01=1
  b11=1
  b02= 0.5
  b12= 1
  r01=2
  r11= 2
  r02=2
```

```
  r12=1
  L=c()
  val=1
# use function "multinom" in package "nnet"
  while (val>1e-04){
    z_star=ntest(x,y,b01,b11,b02,b12,r01,r11,r02,r12)
    data=data.frame(cbind(z_star,simu))
    colnames(data) <- c("z1","z2","z3","x","y")
    model1<multinom(cbind(z3,z1,z2)~x, data)
    model2<glm(y~x, family=poisson, weights=z1, data)
    model3<glm(y~x, family=poisson, weights=z2, data)
    b01=summary(model1)$coefficients[1,1]
    b11=summary(model1)$coefficients[1,2]
    b02=summary(model1)$coefficients[2,1]
    b12=summary(model1)$coefficients[2,2]
    r01=summary(model2)$coefficients[1,1]
    r11= summary(model2)$coefficients[2,1]
    r02=summary(model3)$coefficients[1,1]
    r12= summary(model3)$coefficients[2,1]
    L=c(L,lic(x,y,b01,b11,b02,b12,r01,r11,r02,r12))
    val=AAC(L)
  }
  a=c(b01,b11,b02,b12,r01,r11,r02,r12)
  outcome=rbind(outcome,a)
}
apply(outcome,2,mean)
apply(outcome,2, function(x) sd(x)/sqrt(length(x)))
A=matrix(c(b1,b2,b3,b4,r1,r2,r3,r4),ncol=8, nrow=100, byrow=T)
D=outcome A
apply(D,2, function(x) mean(x^2))
```

B.1.1 Model selection

```
library(LaplacesDemon)
library(nnet)
# simulate data
b1= 0.2519125
b2=0.6086653
b3= 1.6404869
b4=1.8749385

r1=1.8032379
r2=0.5288724
r3=3.8677230
r4=1.2680755
n=100
x=rbern(n,0.5)
p1=exp(b1+b2*x)/(1+exp(b1+b2*x)+exp(b3+b4*x))
p2=exp(b3+b4*x)/(1+exp(b1+b2*x)+exp(b3+b4*x))
p3=1/(1+exp(b1+b2*x)+exp(b3+b4*x))
pi=cbind(p1,p2,p3)
t1=exp(r1+r2*x)
t2=exp(r3+r4*x)
t3=rep(0,n)
t=cbind(t1,t2,t3)
# use function "multinom" in package "nnet"
z=t(apply(pi,1,function(w) rmultinom(1,1,w)))
z_true= apply(z, 1, function(z) { z=(1:length(z))[z==max(z)]; return(z[1]) })
com1=length(which(z_true == 1))
com1
com2=length(which(z_true == 2))
com2
```

```
com3=length(which(z_true == 3))
com3
T=c()
y=c()
for (i in 1:n){
  T=rbind(T,c(rpois(1,t[i,1]),rpois(1,t[i,2]),0))
  y=rbind(y,z[i,]%*%T[i,])

}
simu=cbind(x,y)

# Initial value: c(b01, b11, r01, r11)
# EM algorithm:
# E step:
ntest=function(x,y,b01,b11,r01,r11){
  # p and t are vectors
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x))
  p2=1 p1
  t1=exp(r01+r11*x)
  ztest=matrix(ncol=2,nrow=length(x))
  for (i in 1:length(x)){
    if (y[i]==0){
      z1=(p1[i]*exp( t1[i]))/(p1[i]*exp( t1[i])+p2[i])
      ztest[i,]=c(z1,1 z1)
    } else if (y[i]>0) {
      ztest[i,]=c(1,0)
    }
  }
  return(ztest)
}

```

```
# likelihood function:
lic=function(x,y,b01,b11,r01,r11){
  l=c()
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x))
  p2=1 p1
  t1=exp(r01+r11*x)

for (i in 1:length(x)){
    if (y[i]>0) {
      l[i]=log(1*dpois(y[i],t1[i]))
    } else {
      l[i]=log(p1[i]*exp( t1[i])+p2[i])
    }
  }
  L=sum(l)
  c(L)
}

#stop criteria function:
AAC=function(L){
  if (length(L)<4){
    val=1
    return(val)
  } else {
    # b is the length of L
    b=length(L)
    Ck_1=(L[b] L[b 1])/(L[b 1] L[b 2])
    la_k1=L[b 1]+(L[b] L[b 1])/(1 Ck_1)
    Ck=(L[b 1] L[b 2])/(L[b 2] L[b 3])
    la_k=L[b 2]+(L[b 1] L[b 2])/(1 Ck)
    val=abs(la_k1 la_k)
```

```
92      if(is.nan(val)) val=0
93      return(val)
94    }
95  }
96
97  L=c()
98  val=1
99  while (val>1e 04){
100   z_star=ntest(x,y,b01,b11,r01,r11)
101   z_test = apply(z_star, 1, function(z) { z=(1:length(z))[z==max(z)]; return(z
        [1]) })
102   data=data.frame(cbind(z_star,simu))
103   colnames(data) < c("z1","z2","x","y")
104   model1 < glm(cbind(z1,z2)~x, data, family=quasibinomial)
105   model2 < glm(y~x, family=poisson, weights=z1, data)
106   b01=model1$coefficients[1]
107   b11=model1$coefficients[2]
108   r01=model2$coefficients[1]
109   r11=model2$coefficients[2]
110   L=c(L,lic(x,y,b01,b11,r01,r11))
111   print(L)
112   val=AAC(L)
113 }
114 L
115 BIC1= 2*L+4*log(113)
116 BIC1
117 ##############################################################################
118 # Initial value: c(b01,b11,b02,b12,r01,r11,r02,r12)
119 # EM algorithm:
120 # E step:
121 ntest=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
```

```
  # p and t are vectors
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p3=1 p1 p2
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  ztest=matrix(ncol=3,nrow=length(x))
  for (i in 1:length(x)){
    if (y[i]==0){
      z1=(p1[i]*exp( t1[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
      z2=(p2[i]*exp( t2[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
      ztest[i,]=c(z1,z2,1 z1 z2)
    } else if (y[i]>0) {
      z1=(p1[i]*dpois(y[i],t1[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i]))
      z2=(p2[i]*dpois(y[i],t2[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i]))
      ztest[i,]=c(z1,z2,0)
    }
  }
  return(ztest)
}

# likelihood function:
lic=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
  l=c()
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
  p3=1 p1 p2
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
```

```
  for (i in 1:length(x)){
    if (y[i]>0) {
      l[i]=log(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],t2[i]))
    } else {
      l[i]=log(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i] )
    }
  }
  L=sum(l)
  c(L)
}

#stop criteria function:
AAC=function(L){
  if (length(L)<4){
    val=1
    return(val)
  } else {
    # b is the length of L
    b=length(L)
    Ck_1=(L[b] L[b 1])/(L[b 1] L[b 2])
    la_k1=L[b 1]+(L[b] L[b 1])/(1 Ck_1)
    Ck=(L[b 1] L[b 2])/(L[b 2] L[b 3])
    la_k=L[b 2]+(L[b 1] L[b 2])/(1 Ck)
    val=abs(la_k1 la_k)
    if(is.nan(val)) val=0
    return(val)
  }
}

L=c()
val=1
```

```
# use function "multinom" in package "nnet"
while (val>1e 04){
  z_star=ntest(x,y,b01,b11,b02,b12,r01,r11,r02,r12)
  z_test = apply(z_star, 1, function(z) { z=(1:length(z))[z==max(z)]; return(z
    [1]) })
  data=data.frame(cbind(z_star,simu))
  colnames(data) < c("z1","z2","z3","x","y")
  model1 < multinom(cbind(z3,z1,z2)~x, data)
  model2 < glm(y~x, family=poisson, weights=z1, data)
  model3 < glm(y~x, family=poisson, weights=z2, data)
  b01=summary(model1)$coefficients[1,1]
  b11=summary(model1)$coefficients[1,2]
  b02=summary(model1)$coefficients[2,1]
  b12=summary(model1)$coefficients[2,2]
  r01=summary(model2)$coefficients[1,1]
  r11= summary(model2)$coefficients[2,1]
  r02=summary(model3)$coefficients[1,1]
  r12= summary(model3)$coefficients[2,1]
  L=c(L,lic(x,y,b01,b11,b02,b12,r01,r11,r02,r12))
  print(L)
  #plot(L)
  val=AAC(L)
}
BIC2= 2*L+8*log(113)
BIC2
#############################################################################
#initial value: c(b01, b11, b02, b12, b03, b13,r01,r11,r02,r12,r03,r13)
# EM algorithm:
# E step:
ntest=function(x,y,b01,b11,b02,b12,b03,b13,r01,r11,r02,r12,r03,r13){
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
```

```
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
  p3=exp(b03+b13*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
  p4=1 p1  p2  p3
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  t3=exp(r03+r13*x)
  ztest=matrix(ncol=4,nrow=length(x))
  for (i in 1:length(x)){
    if (y[i]==0){
      z1=(p1[i]*exp( t1[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i]*exp(
    t3[i])+p4[i])
      z2=(p2[i]*exp( t2[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i]*exp(
    t3[i])+p4[i])
      z3=(p3[i]*exp( t3[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i]*exp(
    t3[i])+p4[i])
      ztest[i,]=c(z1,z2,z3,1 z1  z2  z3)
    } else if (y[i]>0) {
      z1=(p1[i]*dpois(y[i],t1[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i])+p3[i]*dpois(y[i],t3[i]))
      z2=(p2[i]*dpois(y[i],t2[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i])+p3[i]*dpois(y[i],t3[i]))
      z3=(p3[i]*dpois(y[i],t3[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],
    t2[i])+p3[i]*dpois(y[i],t3[i]))
      ztest[i,]=c(z1,z2,z3,0)
    }
  }
  return(ztest)
}

# likelihood function:
lic=function(x,y,b01,b11,b02,b12,b03,b13,r01,r11,r02,r12,r03,r13){
```

```
  l=c()
  p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
  p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
  p3=exp(b03+b13*x)/(1+exp(b01+b11*x)+exp(b02+b12*x)+exp(b03+b13*x))
  p4=1 p1  p2  p3
  t1=exp(r01+r11*x)
  t2=exp(r02+r12*x)
  t3=exp(r03+r13*x)
  for (i in 1:length(x)){
    if (y[i]>0) {
      l[i]=log(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],t2[i])+p3[i]*dpois(y[i
    ],t3[i]))
    } else {
      l[i]=log(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i]*exp( t3[i])+p4[i])
    }
  }
  L=sum(l)
  c(L)
}

#stop criteria function:
AAC=function(L){
  if (length(L)<4){
    val=1
    return(val)
  } else {
    # b is the length of L
    b=length(L)
    Ck_1=(L[b] L[b 1])/(L[b 1] L[b 2])
    la_k1=L[b 1]+(L[b] L[b 1])/(1 Ck_1)
    Ck=(L[b 1] L[b 2])/(L[b 2] L[b 3])
```

```
    la_k=L[b 2]+(L[b 1] L[b 2]) /(1 Ck)
    val=abs(la_k1 la_k)
    if(is.nan(val)) val=0
    return(val)
  }
}

L=c()
val=1
k=0
# use function "multinom" in package "nnet"
while (val>1e 06){
  k=k+1
  z_star=ntest(x,y,b01,b11,b02,b12,b03,b13,r01,r11,r02,r12,r03,r13)
  data=data.frame(cbind(z_star,simu))
  colnames(data) < c("z1","z2","z3","z4","x","y")
  model1 < multinom(cbind(z4,z1,z2,z3)~x, data)
  model2 < glm(y~x, family=poisson, weights=z1, data)
  model3 < glm(y~x, family=poisson, weights=z2, data)
  model4 < glm(y~x, family=poisson, weights=z3, data)
  b01=summary(model1)$coefficients[1,1]
  b11=summary(model1)$coefficients[1,2]
  b02=summary(model1)$coefficients[2,1]
  b12=summary(model1)$coefficients[2,2]
  b03=summary(model1)$coefficients[3,1]
  b13=summary(model1)$coefficients[3,2]
  r01=summary(model2)$coefficients[1,1]
  r11= summary(model2)$coefficients[2,1]
  r02=summary(model3)$coefficients[1,1]
  r12= summary(model3)$coefficients[2,1]
  r03= summary(model4)$coefficients[1,1]
```

```
  r13= summary(model4)$coefficients[2,1]
  L=c(L,lic(x,y,b01,b11,b02,b12,b03,b13,r01,r11,r02,r12,r03,r13))
  print(L)
  val=AAC(L)
}
BIC3= 2*L+12*log(113)
BIC3
```

B.1.2 Mapping

```
# mapping
library(LaplacesDemon)
library(nnet)
# simulate data
b1=0.904
b2= 0.170
b3= 0.514
b4=0.737
r1=3.782
r2= 1.102
r3=5.131
r4= 0.720
n=100
x=rbern(n,0.5)
p1=exp(b1+b2*x)/(1+exp(b1+b2*x)+exp(b3+b4*x))
p2=exp(b3+b4*x)/(1+exp(b1+b2*x)+exp(b3+b4*x))
p3=1/(1+exp(b1+b2*x)+exp(b3+b4*x))
pi=cbind(p1,p2,p3)
t1=exp(r1+r2*x)
t2=exp(r3+r4*x)
t3=rep(0,n)
```

```
t=cbind(t1,t2,t3)
# use function "multinom" in package "nnet"
z=t(apply(pi,1,function(w) rmultinom(1,1,w)))
z_true= apply(z, 1, function(z) { z=(1:length(z))[z==max(z)]; return(z[1]) })
com1=length(which(z_true == 1))
com1
com2=length(which(z_true == 2))
com2
com3=length(which(z_true == 3))
com3
T=c()
y=c()
for (i in 1:n){
  T=rbind(T,c(rpois(1,t[i,1]),rpois(1,t[i,2]),0))
  y=rbind(y,z[i,]%*%T[i,])

}
simu=cbind(x,y)
b01=1
b11=0.5
b02=1
b12=1
r01=2
r11=2
r02=2
r12=1
# EM algorithm:
# E step:
  ntest=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
    # p and t are vectors
    p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
```

```
53      p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
54      p3=1 p1  p2
55      t1=exp(r01+r11*x)
56      t2=exp(r02+r12*x)
57      ztest=matrix(ncol=3,nrow=length(x))
58      for (i in 1:length(x)){
59        if (y[i]==0){
60          z1=(p1[i]*exp( t1[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
61          z2=(p2[i]*exp( t2[i]))/(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i])
62          ztest[i,]=c(z1,z2,1 z1  z2)
63        } else if (y[i]>0) {
64          z1=(p1[i]*dpois(y[i],t1[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i
    ],t2[i]))
65          z2=(p2[i]*dpois(y[i],t2[i]))/(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i
    ],t2[i]))
66          ztest[i,]=c(z1,z2,0)
67        }
68      }
69      return(ztest)
70    }
71
72  # likelihood function:
73  lic=function(x,y,b01,b11,b02,b12,r01,r11,r02,r12){
74    l=c()
75    p1=exp(b01+b11*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
76    p2=exp(b02+b12*x)/(1+exp(b01+b11*x)+exp(b02+b12*x))
77    p3=1 p1  p2
78    t1=exp(r01+r11*x)
79    t2=exp(r02+r12*x)
80    for (i in 1:length(x)){
81      if (y[i]>0) {
```

```
      l[i]=log(p1[i]*dpois(y[i],t1[i])+p2[i]*dpois(y[i],t2[i]))
    } else {
      l[i]=log(p1[i]*exp( t1[i])+p2[i]*exp( t2[i])+p3[i] )
    }
  }
  L=sum(l)
  c(L)
}

#stop criteria function:
AAC=function(L){
  if (length(L)<4){
    val=1
    return(val)
  } else {
    # b is the length of L
    b=length(L)
    Ck_1=(L[b] L[b 1])/(L[b 1] L[b 2])
    la_k1=L[b 1]+(L[b] L[b 1])/(1 Ck_1)
    Ck=(L[b 1] L[b 2])/(L[b 2] L[b 3])
    la_k=L[b 2]+(L[b 1] L[b 2])/(1 Ck)
    val=abs(la_k1 la_k)
    if(is.nan(val)) val=0
    return(val)
  }
}

L=c()
val=1
# use function "multinom" in package "nnet"
while (val>1e 04){
```

```
  z_star=ntest(x,y,b01,b11,b02,b12,r01,r11,r02,r12)
  z_test = apply(z_star, 1, function(z) { z=(1:length(z))[z==max(z)]; return(z
    [1]) })
  data=data.frame(cbind(z_star,simu))
  colnames(data) < c("z1","z2","z3","x","y")
  model1 < multinom(cbind(z3,z1,z2)~x, data)
  model2 < glm(y~x, family=poisson, weights=z1, data)
  model3 < glm(y~x, family=poisson, weights=z2, data)
  b01=summary(model1)$coefficients[1,1]
  b11=summary(model1)$coefficients[1,2]
  b02=summary(model1)$coefficients[2,1]
  b12=summary(model1)$coefficients[2,2]
  r01=summary(model2)$coefficients[1,1]
  r11= summary(model2)$coefficients[2,1]
  r02=summary(model3)$coefficients[1,1]
  r12= summary(model3)$coefficients[2,1]
  L=c(L,lic(x,y,b01,b11,b02,b12,r01,r11,r02,r12))
  #print(L)
  #plot(L)
  val=AAC(L)
}
testz=z_test
com1_new=length(which(z_test == 1))
com1_new
com2_new=length(which(z_test == 2))
com2_new
com3_new=length(which(z_test == 3))
com3_new
delta=z_test  z_true
delta
z_test
```

```
143 z_true
```

B.2 Calculate ARI

```
1  library(mclust)
2  a < rep(1, 16)
3  b < rep(2, 15)
4  c < rep(3, 23)
5  d < rep(4, 46)
6  m < c(a,b,c,d)
7  a1 < rep(1, 16)
8  b1 < rep(2, 14)
9  c1 < rep(3, 1)
10 d1 < rep(3, 23)
11 e1 < rep(4, 46)
12 n < c(a1,b1,c1,d1,e1)
13 # use function "adjustedRandIndex" in package "mclust"
14 adjustedRandIndex(m, n)
```